People Detection based on Points Tracked by an Omnidirectional Camera and Interaction Distance for Service Robots System(Dissertation_全文)

AUTHOR(S):

Tasaki, Tsuyoshi

People Detection based on Points Tracked by an Omnidirectional Camera and Interaction Distance for Service Robots System

Tsuyoshi TASAKI

Abstract

A robotic system is expected to increase visitors for big facilities such as big shopping stores. The recent robotic system often has two kinds of service robots, interaction robots and mobile robots. The interaction robots attract many visitors simultaneously at a particular position in the facilities. The mobile robots provide services to each visitor wherever he/she wants to go. For example, the mobile robots navigate visitors or transport their baggage.

It is important for both interaction robots and mobile robots to detect people and objects. The interaction robot has to detect an interaction partner from among multiple people in order to interact with multiple people simultaneously. The mobile robots have to detect all moving people around the robot in order to avoid them smoothly and have to detect landmark objects in order to localize themselves.

Most interaction robots have multiple sensors in order to realize rich functions. For example, they integrate multiple sensors in order to detect and interact with people. However, many works assume the robot interacts with a person and they have not dealt with how the robot selects an interaction partner from among multiple people based on the plausible criteria.

The robotic system needs a lot of mobile robots in order to provide services to each visitor. Therefore, the mobile robots should be low-cost. In order to develop low-cost mobile robots, some works detect all objects around the robots by a single sensor, such as omnidirectional camera. However, it is difficult to classify objects as dynamic or not while robots move, because of distortion of the omnidirectional image. The distortion also makes it difficult to detect landmarks fast.

We deal with following three problems related to detecting people and objects for a robotic system.

1. An interaction robot that has multiple sensors detects interaction partners based on the plausible criteria in order to interact with multiple people friendly.

2. A mobile robot that has only one omnidirectional camera classifies all objects around the robot as dynamic obstacles or not while the robot moves.

3. A mobile robot that has only one omnidirectional camera detects landmarks fast for the self-localization.

In order to solve the first problem, we focus on the distance between the interaction robot and each person. We have developed a method to select an interaction partner based on the degree of friendliness mapped onto the " space " which is divided by the interaction distance and the effective distances of each sensor.

In order to solve the second and the third problem, we focus on local feature points which can be detected even in the distorted omnidirectional images. We have developed a new method that focuses on floor boundary points where the robot can measure the distance from itself by an omnidirectional camera. Our robot classifies a floor boundary point as a dynamic obstacle when its movement is different from the robot's movement.

We have also developed a new method that uses tracked local invariant feature points that are detected by both "fast" tracking method and "slow" Speed Up Robust Features (SURF) method.

Solving these three problems, we aim to develop the following robot system.

1. The system includes the interaction robot that can select friendly and plausible action while the robot interacts with multiple people.

2. The system includes the mobile robots that can classify dynamic obstacles or not while robot moves.

3. The system includes the mobile robots that can localize themselves fast and accurately.

In Chapter 1, we describe the motivations for and goal of this study. We then briefly describe the issues mentioned above and the approaches. We describe a robot system which has an interaction robot and mobile robots. We define a condition where our robots work.

In Chapter 2, we review various works in related fields. The review mainly covers works on the human robot interaction, the obstacle detection and the landmark detection. We point out the problems of previous works on the interaction between a robot and multiple

people. We explain that it is difficult for previous methods to detect dynamic obstacles and landmarks fast while the robot moves by using a single omnidirectional camera.

Chapter 3 describes the new method detecting an interaction partner from among multiple people based on the interaction distance. Proxemics, which is a social psychology theory, says that two people interact at an appropriate interaction distance from one another based on their relationship. In this theory, the interaction distance can be classified into roughly four groups. Moreover, effective distance for robot's functions can correspond to the interaction distance effectively. Therefore, we divide the space around the robot based on the interaction distance. Our robot maps friendliness onto the divided spaces and detect interaction partner based on the friendliness.

In Chapter 4, we evaluate the interaction partner detection. Our humanoid robot, SIG2 which the proposed method is implemented into, interacted with about 30 visitors. The results obtained using questionnaires after interaction show that the actions of SIG2 is easy to understand even when it interacted with multiple people at the same time and that SIG2 behaved in a friendly manner.

Chapter 5 describes the obstacle detection method based on the floor boundary points. In order to locate obstacles, we regard floor boundary points where robots can measure the distance from the robot by one omnidirectional camera as obstacles. Tracking them, we can classify obstacles by comparing the movement of each tracked point with odometry data. Moreover, our method changes a threshold to detect the points based on the result of comparing in order to enhance detection.

Chapter 6 describes "tracked local invariant feature points" that are regarded as landmarks. These landmarks can be tracked and do not change for a "long" time. In a landmark selection phase, robots detect the feature points by using both a fast tracking method and a slow SURF method. After detection, robots select landmarks from among detected feature points by using Support Vector Machine (SVM) trained by feature vectors based on observation positions. In a self-localization phase, robots detect landmarks while switching detection methods dynamically based on a tracking error criterion that is calculated easily even in the uncalibrated omnidirectional image.

In Chapter 7, we performed experiments in a mock shopping store by using a navigation robot ApriTau^{TM} that had an omnidirectional camera on its top. The results showed that a classification ratio of our dynamic obstacles detection was 4.0 times higher than that of the previous method. Our method could localize 2.9 times faster than the

previous method could. Moreover, our method reduced the error of localization to 23.6% of a previous method.

In Chapter 8, we demonstrate the robot navigation service and the robot transportation service. We discuss the major contributions of this study towards related research fields, the human robot interaction, the obstacle detection and the landmark detection. We also discuss remaining issues and future directions of our work.

Finally, we present the conclusion of this thesis in Chapter 9.

和文題目：
「サービスロボットのための全方位カメラによるトラッキング可能な特徴点とインタラクション距離情報を用いた人物検出」

田崎豪

内容梗概

集客力向上のため、大規模施設でのロボットサービスが期待されている。特に、大規模な店舗では、イベントホールのような特定の場所で、一度にたくさんの客を相手にするインタラクションロボットと、個々の客を相手に案内や手荷物運搬のサービスを行う移動ロボットの二種類のサービスロボットが多く開発されている。

二種類のロボットが提供するサービスは異なるが、どちらのロボットにおいても、人や物を検出する機能は重要である。インタラクションロボットは、多くの客と同時にインタラクションできるよう、複数人の中から特にインタラクションすべき人を検出することが重要になる。移動ロボットにおいては、安全確実な移動のために、障害物や目印となる物体を検出する必要がある。

インタラクションロボットの多くが、人とインタラクションを行うという高機能の実現のため、複数のセンサを持つ。しかし従来の研究では、複数人の中から、個々の人にとってわかりやすく親しみやすい基準を持って、人を選択することが行われていない。

移動ロボットについては、個々の客に対応するため、価格や設定などのコストが安い、大量生産可能なロボットが求められている。安全確実な移動を実現しつつ、コストを削減するため、一つのセンサだけで、全周囲の障害物を検出し、目印の検出を行う必要がある。従来、一つのセンサで、全周囲の物体を検出する手段として、全方位カメラが用いられてきた。全方位カメラでは、全周囲の静止障害物を検出する研究が多いが、画像の歪みが大きいため、移動ロボットの経路生成時に特に注意すべき、移動障害物と静止障害物を、ロボット移動中に区別して検出する手法はあまり開発されていない。また、歪みのため、全周囲の目印を高速に検出することは難しい。

本研究では、サービスロボットのための、人と物体の検出に関して、以下の３つの課題に取り組む。

1. 複数人インタラクション時でも個々の人にわかりやすく親しみやすい選択基準を持ったインタラクションパートナー検出

2. 一つの全方位カメラを用いた移動ロボット移動中の全周囲移動障害物検出

3. 全方位カメラによる高速な目印検出を用いた自己位置推定

１つ目の課題の解決のため、我々は、ロボットとインタラクションを行うそれぞれの人とのインタラクション距離に着目した。人間同士の親密性を示すインタラクション距離の分類と、ロボット搭載センサの使用可能距離に基づきロボットインタラクション空間を分割する。分割された個々の空間に、人の存在度合いと、インタラクション中の刺激の快不快をもとにした親密度をマッピングすることで、人にわかりやすい選択基準で、インタラクションパートナーを検出する。

２つ目と３つ目の課題を解決するため、我々は、全方位画像中でも歪みの影響を受けにくい局所特徴点に着目した。全方位カメラ画像中の床面と障害物の間を床際点で表し、個々の床際点で障害物が移動しているか否かを識別し、ロボット移動中であっても、全周囲の移動障害物を検出する。さらに、目印検出には、全方位画像中でも低速ではあるが、特徴点対応が可能なSpeed Up Robust Features (SURF) と、高速に動作するトラッキング手法を統合したトラッキング可能な回転スケール不変特徴点を新たに開発した。SURFの回転スケール不変特性と、トラッキングの高速性の利点を生かし、高速に目印を検出し、正確に自己位置推定を行う。

課題解決により、以下の２種類のサービスロボットの開発を目的とする。

1. 複数人とのインタラクションを行っても個々の人に親しみやすくわかりやすい行動選択が行えるインタラクションロボット

2. ただ一つのセンサだけを使用し、移動中であっても、特に移動する人を障害物として検出し、高速かつ正確に自己位置を推定できる移動ロボット

以下第１章では、我々が目指しているインタラクションロボットと移動ロボットについて、詳細を述べる。また、導入を予定している大規模店舗の環境条件を前提条件として明記する。我々が３つの課題に取り組むことを述べ、アプローチを述べる。

第２章では、現在のインタラクションロボットにおける人検出の研究と、全方位カメラによる障害物検出、自己位置推定の研究を紹介する。インタラクションについては、現

在のインタラクションロボットにおける、複数人インタラクションに対する問題点を指摘する。さらに、全方位カメラを一つだけ用いて、移動中に移動障害物を検出することと、高速な目印検出が困難であることを現在の研究をもとに説明する。

第３章では、インタラクション距離に基づく、インタラクションパートナー検出手法について紹介する。近接学と呼ばれる社会心理学の分野では、人と人との親密性が、インタラクション時の距離に基づいて分類できるといわれている。さらに、ロボットのセンサの使用可能距離が、近接学で設定される３つの距離と対応がとれるということに着目し、ロボット周囲の空間を親密度空間マップと呼ばれるインタラクションを行う解像度に適した空間に分割する。分割された空間上で、センサ情報を統合し、同時にインタラクションを行う複数の人それぞれにとって、わかりやすく親しみやすい人検出を行う手法について説明する。

第４章では、親密度空間マップをもとにした、インタラクションの客観的、主観的評価を行う。評価のために使用した、我々のインタラクションロボット SIG2 を紹介し、SIG2 と複数の人とのインタラクション実験を行う。実験の結果、インタラクションを観測していた、観測者がインタラクションすべきであると指示した人と、SIG2 がインタラクションを行ったインタラクションパートナーが一致していることを明らかにした。さらに、インタラクションを行った実験協力者自身にも SD 法に基づくアンケートに答えてもらい、わかりやすく、親しみやすいインタラクションが実現できていることを確認した。

第５章では、床際点を用いた移動中の移動障害物検出手法について紹介する。床色を Ward 法によって学習し、床面を検出する。床面と床面以外の境界を床際点として検出し、床際点を時系列でトラッキングする。検出誤差を減らすため、床際点検出の閾値を弱くして、短期的に行うトラッキングと、閾値を強くして、長期的に行うトラッキングの二種類で床際点をトラッキングした。二種類のトラッキングの結果、ロボットの移動と異なる移動を行った床際点を移動障害物として検出する。

第６章では、トラッキング可能回転スケール不変特徴点を紹介し、トラッキング可能な回転スケール不変特徴点を利用した高速な自己位置推定手法について述べる。全方位画像中でも頑健に検出可能ではあるが、検出に時間がかかる SURF と、全方位画像中でも高速に動作するトラッキングを併用する。時間をかけてもよい目印学習時には、施設内全体を全方位カメラで撮像し、撮像したすべての画像について SURF 特徴点を検出する。得られた特徴点のうち、トラッキングを長時間で来ていたものについて、トラッキング可能な回転スケール不変特徴点として検出する。

第７章では、移動障害物検出と、高速目印検出について、我々が開発した移動ロボット ApriTau^{TM} を用いて、実験を行った。実験の結果、移動障害物検出は、F 値が従来の

4.0 倍高くなり、安全な移動を行える可能性があることを示した。さらに、目印検出では、従来手法より誤差を 23.6%に抑制でき、2.9 倍高速にロボット位置推定が行えることを確認した。第８章では、サービスへの応用例として、案内サービスと手荷物運搬サービスを行った様子について述べる。また、各実験で得られた結果について考察を述べ、残された課題について議論する。最後に第９章でまとめを行う。

Acknowledgments

This work was carried out at Okuno Laboratory, Graduate School of Informatics, Kyoto University. I would like to express my gratitude to all the people who have supported this work.

First and foremost, I would like to thank Professor Hiroshi G. Okuno for his thoughtful supervision throughout my studies. My theory and approach for work were based on his advice and idea. His wide knowledge gave me new idea. His power of growing researchers gave me a way to researcher.

Another person who had greatly supported my work was Professor Tetsuya Ogata. His expertise in the robotics field has created the foundation of the approach of my work. His discussion to a result gave me a new viewpoint of research and I could continue new work.

I would like to thank Professor Tatsuya Kawahara, Professor Yuichi Nakamura and Professor Atsushi Igarashi for a prompt and kind review.

This work has been supported by many of my colleagues at Okuno Laboratory. Associate Professor Kazunori Komatani taught me a difficulty of research. Mr. Mitsuhiko Toda, Mr. Shohei Matsumoto and Mr. Hayato Ohba helped my experiments. Dr. Shun Nishide and Dr. Katsutoshi Itoyama gave me a friendly atmosphere at the laboratory. Mr. Takeshi Mizumoto and Mr. Takuma Otsuka gave my research a lot of remarks.

I would also like to express my gratitude to Professor Nobuto Matsuhira. He gave me a work for robots at Toshiba Corporation. His broad connection of robot workers at Toshiba gave me a chance of big work.

This work has been also supported by my colleagues at Toshiba Corporation and Toshiba Tec Corporation. Dr. Fumio Ozaki conducted a project of a robot and taught me how to work. Mr. Hirokazu Sato made an electric system of a robot which I used. Mr. Hideki Ogawa introduced me robots of Toshiba Corporation. Dr. Junji Oaki taught me research of Toshiba. Ms. Junko Hirokawa made a mechanical design of a robot. Mr.

Takafumi Sonoura made a control system of a robot. Dr. Seiji Tokura integrated a robot system. Dr. Ryohei Orihara, Dr. Miwako Doi, Mr. Kazuki Taira and Dr. Daisuke Yamamoto supported me for doctor course. Mr. Yoshifumi Tanabe, Mr. Masahiko Sano, Mr. Yasuhiro Inagaki, Mr. Tsuyoshi Takanose and Ms. Akiko Numata gave me an experimental environment.

Finally, I would like to thank my family for their understanding and support during my studies. My father Masafumi and mother Yuko raised me with my sister Nao. My wife Tomoko and daughter Nodoka gave me happiness when I was tired. I could work and study through their help.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter briefly describes the motivation, goal, issues, and approaches of this thesis.

1.1 Motivation

Many robots which work around people have been developed since specially 1990. The famous examples of such robots are Honda ASIMO[1][2], Sony QRIO[3], Toshiba ApriAlpha^{TM}[4], and ApriAttenda^{TM}[5]. In around 2000, they were demonstrated only under fixed situations. Recently, there are some robots which work at big facilities where people walk, which are not fixed situations[6][7].

In order to increase visitors, two kinds of robots work at big facilities. One is an interaction robot which attracts a lot of people at a particular position as like event halls[8]. The other is a mobile robot which transports visitors' baggage or navigates visitors[9]. Many works have developed such robots. Robovie interacts with a person by using a lot of sensors in order to call many visitors at a big shopping store[10]. Panasonic porter robot[11], Murata porter robot[12], Korean robot[13] and so on transport baggage of visitors at big shopping stores or hospitals, airports.

The robotic system at the big facilities consists of such two kinds of robots. However, the features of them are different. The interaction robot interacts with more people at the same time than mobile robots do in order to call many people. The number of the interaction robot is less than that of the mobile robots because the interaction robot works at a particular position and the mobile robots work at various positions. Certainly, both robots are expected to be low-cost, but the cost of the mobile robot should be lower than that of the interaction robot because the system needs a lot of mobile robots for each visitor. In order to develop low-cost mobile robots, they are expected to use a single

sensor which can acquire all information around the robots for the obstacle classification, the self-localization. On the other hand, most interaction robots are allowed to use various sensors which are similar to human's eye, ear, and skin for interaction.

Although the features of both robots are different, the function of recognizing environments around them is important. Specially, detecting people and objects around them is one of the most important functions of all robots' functions because they should interact with people friendly and move safely. Therefore, we deal with the function of detecting people and objects around robots.

1.2 Goal and Issues

In this thesis, we deal with detecting people and objects around the robot at the big facilities. Among big facilities, a number of the big shopping store is the largest. Therefore, we supply our interaction robot and mobile robots with the big shopping stores. Our interaction robots work at the event halls in the store. The illumination condition often changes at the halls. The hall announcements are broadcasted regularly. Our mobile robots move on a shopping floor which has a lot of shelves for commodities. Main stable obstacles are shelves and walls. They have a boundary whose color is different from the floor color. The floor has some figures. However, illumination condition does not change. It has more than 2.0 [m] paths. There are a few people on the path.

In the big shopping stores, the interaction robot is expected to call many people there. Therefore, it should interact with multiple people simultaneously. The goal of detecting people and objects on the interaction robot is (1) to interact with people by friendly manner.

On the other hand, the mobile robots should move safely, smoothly and precisely. In order to move safely and smoothly, it is important for them to localize and classify all obstacles around the robot as stable or not while the robot moves. In order to move precisely, it is important to localize the mobile robot itself fast and precisely. These clasificaion and localization should be implemented by a single sensor for cost reduction. An omnidirectional camera is one of the most appropriate sensors. Therefore, we use it. Here, the goals of detecting people and objects on the mobile robots are (2) to localize and classify all obstacles around robots as stable or not while they move, and (3) localize themselves fast and precisely by using only one omnidirectional camera. We achieve these

three goals on each robot. Considering as mentioned above, we list the features about the detection for the interaction robot and the mobile robots in Table 1.1. The features of those robots are different. Therefore, we use two kinds of robots, an interaction robot and mobile robots. In our system, the interaction robot cannot transport the baggage and the mobile robots cannot interact with people.

Table 1.1: The features of the detection for the interaction robot and the mobile robots

	Interaction Robot	Mobile Robots
Aim	friendly interaction with multiple people	safe and precise movement
Target	interaction partner	moving obstacles and landmarks
Sensor	We can use various sensors. (cameras, microphones and tactile sensors)	We can use only a sensor. (omnidirectional camera)
Range	front of the robot	all around the robot
Accuracy	low	high
Resolution	low	high

We list issues to achive the goals based on the Table 1.1.

Issue 1 Interaction Robot:

The effective integration of sensors and the design of a plausible criteria for selecting an interaction partner from among multiple people

As shown in Table 1.1, the interaction robot can be equipped with multiple sensors to detect people. Although the interaction does not need high resolution of detecting people, the interaction robot should have a plausible criteria selecting an interaction partner from among multiple people who are interact with it simultaneously by integrating multiple sensors.

Issue 2 Mobile Robot:

Classifying all obstacles around the robot as stable or not by a single omnidirectional camera while it moves

As shown in Table 1.1, the mobile robot uses only one omnidirectional camera. The camera is useful because a single camera can capture all obstacles around the robot while it moves. However, it is difficult to classify them as stable or not, because images captured by the omnidirectional camera is distort and the camera is not stable.

Issue 3 Mobile Robot:

The fast landmark detection by a single omnidirectional camera

As same as Issue 2, the distortion of the omnidirectional camera causes difficulty of the fast landmark detection.

1.3 Overview of our Approaches

We deal with above-mentioned issues through the following approaches:

Solution 1 Interaction Robot:

Designing a criteria selecting an interaction partner and integrating multiple sensors based on the interaction distance

A distance between one person and the other is one of the criteria which show the relationship between two people. Moreover, the distance between a person and the interaction robot is an important criterion of selecting sensors to use. We design divided spaces based on the spatial relationships and sensing spaces which are related to the "interaction distance". Mapping the friendliness onto each divided space, the robot interacts with the highest friendliness spaces selectively. Selecting an interaction partner by using friendliness based on the relationship which are related to the interaction distance. Moreover, the interaction robot selects its action by plausible manner even if it interacts with multiple people simultaneously based on the divided space.

Solution 2 Mobile Robot:

Tracking floor boundary points with changing thresholds dynamically

In order to locate obstacles, we regard floor boundary points where robots can measure the distance from the robot by one omnidirectional camera as obstacles. Tracking them, we can classify obstacles by comparing the movement of each tracked point with odometry data. Using the result of classification, our robots change a threshold to detect the points and enhance the accuracy of classification.

Solution 3 Mobile Robot:

Developing new feature points which can be detected fast at the broad area in the big shopping store

In order to realize the fast self-localization by using a single omnidirectional camera, we use "tracked local invariant feature points" that are regarded as landmarks. These landmarks can be tracked and do not change for a "long" time even in the

omnidirectional images. In a landmark selection phase, robots detect the feature points by using both a fast tracking method and a slow "Speed Up Robust Features (SURF)" method. In a self-localization phase, robots detect landmarks while switching both detection methods dynamically based on a tracking error criterion that is calculated easily even in the uncalibrated omnidirectional image. Once the landmarks are detected by the "slow" SURF method, the robot continues to detect them by the "fast" tracking method. Therefore, we can realize the fast self-localization.

1.4 Outline of the Thesis

This thesis consists of 9 chapters. The organization of this thesis is shown in Figure 1.1.

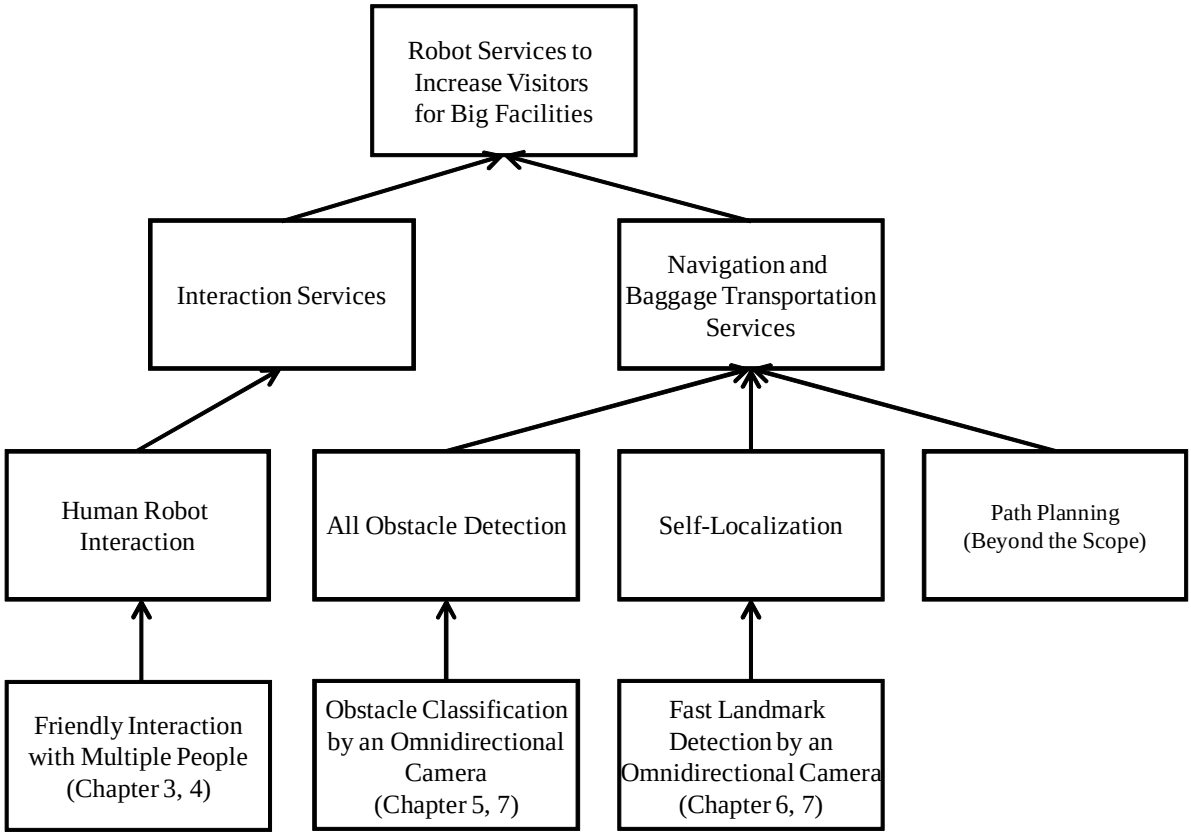


Figure 1.1: The organization of the thesis

Chapter 2 provides a review of the literature in related fields and discusses a positioning of this thesis.

Chapter 3 describes our friendliness space map showing how friendliness is distributed in the space in order to select an interaction partner from among multiple people.

In Chapter 4, we implement the frinedliness space map into our interaction robot SIG2. We also show the result of questionnaires about interaction between a robot and multiple people.

Chapter 5 describes the obstacle classification method based on tracking floor boundary points by using a single omnidirectional camera while a robot moves.

Chapter 6 describes how we detect tracked local invariant feature points by an omnidirectional camera. We explain application to the fast self-localization of a mobile robot.

In Chapter 7, we show our mobile robot ApriTau^{TM} which has an omnidirectional camera. We show an accuracy of our classification method and self-localization method.

In Chapter 8, we discusses the major contribution of this work towards related research fields and the proof of concept related to the robot system at a big shopping store. We describe remaining issues and future directions of our work.

Finally, Chapter 9 concludes the thesis.

Chapter 2

Literature Review

This chapter provides a review of literature related to people detection on human-robot interaction, people detection while robot moves, and fast landmark detection by using camera in order to clarify the positioning of this thesis within related fields.

2.1 People Detection on Multiple Human and Robot Interaction

Many interaction robots are developed, Robisuke [14], SIG [15], AIBO [16], Kismet [17], Wamoeba [18], WE-4 [19], and so on. There are many works detecting people while such a robot interacts with people.

However, research on human-robot interaction thought that it was difficult to detect multiple people by sensors equipped on the robot. Murakita et al. [20] gave a service to convey objects which people wanted to use. It localized people exactly using touch sensors installed over the whole floor, which required installing many devices. The robot used by Kanda et al. [21] was a social robot and its fieldtest had been carries out at an elementary school for two months. The robot successfully understood friendly relationship among students in front of it by tracking their RFID tags attached to student's name tag. Since this field-test needed a precise identification of students, only RFID tag information was utilized.

Putting the sensors on the floor costs and making people have RFID tags are troublesome. We want to localize people by sensors equipped on the robot.

Resently, some works localize people by sensors equipped on the robot.

Miyashita et al. [22] make a robot track a person by intergrating four sensors based on the Markov Montecarlo Method in order to localize a person while the robot interacts

with him/her. However, it regards people whom the robot does not interact with noise in order to enhance detecting the person. The robot cannot interact with multiple people simultaneously. Regarding most people as noise is also problematic for interaction robots.

SIG2 [23] tracks multiple people who are both talking and not talking by integrating visual and auditory localization. It could perform various kinds of visual and auditory scene analyses including face localization and recognition, sound source localization and separation, and automatic speech recognition. Alghough it has various sensory-motor modalities, its behaviors are only passive; it can only track and trun toward a speaker.

Therefore, the works on human-robot interaction has problems for detecting multiple people precisely. However, we think it is unnecessary to localize people in detail for human-robot interaction. People can interact with other people without localizing people in detail. We think it is important to decide how accurate robots localize people.

Moreover, almost of them have dealt with only one-on-one interaction between one robot and one person, there have been quite few discussions on the methodology for the interaction between a robot and multiple people. Considering the yet-to-be developed human support robot, it would be expected that a robot be able to interact effectively with multiple people at the same time. This thesis proposes a design method for such humans-robot interaction.

Robita is a conversation robot that can participate in group discussion [24]. Two people sitting on a chair interact with each other and Robita. During interaction, Robita obtains auditory inputs through a headset microphone worn by each participant. In this sense, its interaction model does not depend on the interaction distance and used the fixed sensory-motor modality. Robita maintains various kinds of information on the blackboard and selects an appropriate module with highest priority [25]. The system architecture is based on inter-module cooperation consisting priprity management, situaged observation, and data exhibition/message dispatch systems. The priority of module is given in advance by a task designer (software developer). Therefore, Robita's behavior is a result of priority based execution control. However, Robita can interact with people very passively. Robita replies to people by same behaviors only when people talk with Robita. Robita does not select the person whom it should interact with. We think it is important for the interaction robot to interact with multiple people actively, because we aim to make multiple people feel positive impression by the human robot interaction even if the robot interacts with multiple people simultaneously.

2.2 Moving People Detection while Robot Moves

In order to detect obstacles around a robot when it moves, many works use distance measurement devices such as the Laser Range Finder (LRF) [26][27] , RGB-D sensor like kinect [28][29], and stereo cameras [30][31][32] However, robots have to be equipped with more than one sensor when they classify all obstacles around them at once by such sensors. Using many sensors is expensive, and calibration is troublesome.

There are many works which detect moving people by a normal single camera while robot moves, although they detect only people in front of the camera. They use a difference between a previous image and a present image with using assumption, geometry or shapes [33]. The methods using assumption assume that there is a person in front of the camera [34]. Others assume movements of people [35]. The methods using geometry use optical flows [36][37]. They regard movements of the optical flows which are different from typical movements of all flows as moving people. Some methods using geometry use the epipolar geometry between a previous image and a present image obtained by robot's movements [38]. Some methods using shapes information use the HOG (Histgram of Oriented Gradients) [39] or coHOG (Co-occurrence Histograms of Oriented Gradients) [40] based on the edge information.

The methods using assumptions do now work well when the assumptions are not true. The methods using geometry or shapes information work well as long as we use a general camera which is not distorted and whose field of view is narrow.

An ominidirectional camera can take images of all obstacles around a robot simultaneously while moving. However, it is difficult to apply previous obstacle classification techniques such as [41][42] to omnidirectional images because the image is distorted and the resolution is low. We can change the omnidirectional images to not distorted general images [43]. Even if we change images, previous techniques do not work well because changed images lose a lot of information. Moreover, classifying obstacles as stable or not by a moving camera is more difficult than classifying by a static camera.

The distortion of the omnidirectional image does not affect colors. Therefore, some works use floor colors in order to detect obstacles, regarding obstacles as objects except floor [44][45][46][47]. However, they do not work well under a dynamic environment changing with robot's movements. They do not consider a change of floor colors by robot's movements. Moreover, they do not classify obstacles as stable or not.

2.3 Object Detection for Self-Localization

For navigation, it is important for robots to localize their own position. Many works deal with self-localization problems. The simplest solution of the problem is to use sensors installed in environements [48] [49]. However, it is difficult to apply them because they spoil a view of the shopping store. The RFIDs are cheap and can localize robots [50]. However, the owner of the big facilities hates constructions. Therefore, we use only a sensor equipped in the robot. Recently, most of works use the LRF equipped in the robots [51][52]. They realize the self-localization, matching a map to data measured using the iterative closest point (ICP) [53][54] algorithm [55]. Self-localization methods based on the LRF data can be performed fast and accurately. However, when robots using the methods do not know their initial position at all, errors occur at airports or big shopping stores. The errors occur because there are many similarly shaped objects such as poles and shelves.

On the other hand, many works use an omnidirectional camera [56][57]. The omnidirectional camera can get not only shape information but also texture information in a wide area around a robot. Therefore, even if there are many similarly shaped objects and a robot does not know its initial position at all, the self-localization succeeds by using the omnidirectional camera.

Some works based on the omnidirectional camera calibrate its mirror parameters [58]. After changing omnidirectional images to non-distorted images, the works apply traditional methods based on general cameras that do not have a wide field of view [59]. These works have an advantage in that they can use many traditional methods. However, changed images lose a lot of information, as mentioned above.

Some works use scale and rotation invariant (Local Invariant: LI) feature points like SIFT [60] or SURF [61] without calibrating mirror parameters. Even if the robot moves and omnidirectional images change, using LI feature points enables the robot to localize its position. However, it performs more slowly and some feature points cannot always be used in the omnidirectional image. Some works dealing with SLAM [62][63] use tracking methods [64] or select feature points that have small errors while making a map [65] in order to perform fast. However, few works related to SLAM focus on feature points that can be tracked for a long time. Therefore, in the case of the previous works, there are many feature points that are not detected from various positions in an area where the

robot moves.

The recent evolution of GPU makes it possible to localize robot's position by using a camera. Parallel Tracking and Mapping (PTAM) [66] is one of the best examples of SLAM using a GPU. However, GPUs consume a lot of electric power, which is not appropriate for mobile robots. Recently, methods such as PTAM not using GPU [67] works in the case of omnidirectonal images. A drawback of PTAM is that it does not have any functions to localize again if robots miss their locations.

2.4 Positioning of this Thesis within Previous People and Object Detection Studies for Robots in the Big Shopping Store

In this section, we discuss the positioning of our work within related people and object detection while the robot interacts with people and while the robot moves. We discuss about the positioning of our work within other works in the following order.

1. Interaction with multiple people.

2. Obstacle classification while robot moves by a single omnidirectional camera.

3. Fast self-localization by an omnidirectional camera.

Works on interaction with multiple people did not have a criterion to select an interaction partner from among multiple people. The works do not detect people by human-like sensors such as eyes, ears, and skins. We have now developed a method for selecting an interaction partner for a robot based on the degree of friendliness as mapped onto the " space ", considering the interaction distance between people and the robot. Recently, the interaction distance has been used for human robot interaction. For example, Satake uses distance to start the interaction [68]. Yamaoka makes a robot keep the interaction distance to introduce other objects [69]. We integrates human-like sensors on the map which consider to the range where the sensors are used on, in order to localize people robustly in various environments, and in order to impress the people interacting with the robot simultaneously as intelligent and friendly.

Works on obstacle classification do not classify all obstacles around the robot as stable or not by a single sensor, while the robot moves. We have developed an original method

of classifying all obstacles around the robot by only one omnidirectional camera while it moves. In order to locate and classify obstacles, we focus on floor boundary points where the robot can measure the distance from itself by only one omnidirectional camera. Our robot classifies a floor boundary point as a dynamic obstacle when its movement is different from the robot's movement.

Works on self-localization by a single omnidirectional camera do not work fast. In order to solve the problems, we focus on the existence of many LI feature points that can be tracked and do not change their feature vectors for a long time. We regard the tracked LI feature points as new landmarks. In a landmark selection phase, we select landmarks from among the tracked LI feature points by using Support Vector Machine (SVM). In a self-localization phase, our robot detects landmarks, switching detection methods dynamically based on a tracking error criterion. The criterion is calculated easily by directions of detected tracked LI feature points even in the uncalibrated omnidirectional image. Because the tracked LI feature points can be tracked for a long time and do not change their feature vectors very much, our robot can detect them from various positions. Moreover, our robot can continue to detect them fast and localize its position even if it does not know their initial position at all.

Chapter 3

Interaction Partner Detection by Spatial Mapping of Friendliness Based on Interaction Distance

This chapter discusses the distance between the robot and people during an interaction and describes our "friendliness space map" showing how "friendliness" is distributed in the space.

3.1 Distance between Robot and People during Interaction

3.1.1 Interaction Distance of People

When people interact with each other, the distance between them is associated with their degree of friendliness. Proxemics [70], which is a social psychology theory, says that two people interact at an appropriate physical distance from one another based on their relationship. In this theory, the interaction distance can be classified into roughly four groups: intimate, personal, social, and public.

- Intimate distance (approx. 50 [cm])

  People can communicate via physical interaction and express strong emotions.

- Personal distance (approx. 50—120 [cm])

  People can talk intimately.

- Social distance (approx. 120—360 [cm])

  People don't know each other well.

- Public distance (approx. 360 [cm] and more)

  People who have no personal relationship with each other can comfortably coexist at this distance.

These distances can be used to set the degree of friendliness between the robot and each person. The distances shown in parentheses are only typical ones. They depend on each person's personality and cultural background.

3.1.2 Effective Distance and Advantages and Disadvantages of Robot's Functions

Since most functions and devices used by a robot are not effective for all distances, we assessed the effective distance for them. We investigated the effective distance of tactile recognition, speech recognition, sound source localization, and face localization, which are implemented into many robots as general functions.

Tactile Recognition

Works on tactile sensing [71][72] verify that the tactile interaction shows emotions and people like tactile interaction with intimate people [73]. Therefore, we use the tactile recognition. The tactile recognition is done using tactile sensors, which are effective when people can touch the robot. The average length of a person's arm is about 70 [cm], so the appropriate distance for tactile recognition is up to 50 [cm]. This distance is similar to the intimate distance.

Speech Recognition

To determine the range for speech recognition, we place a speaker in front of a robot at every 50 [cm] from 50 [cm] to 3 [m] and played 200 words of the ATR phonetically balanced corpus. The results of isolated word recognition using "Julian" [74], general Japanese automatic speech recognition software, are shown in Figure 3.1. Automatic speech recognition was found to be effective up to around 1.5 [m].

Sound Source Localization

A well-known sound source localization function uses the Interaural Phase Difference (IPD) and Interaural Intensity Difference (IID) [75]. We explain the integration method

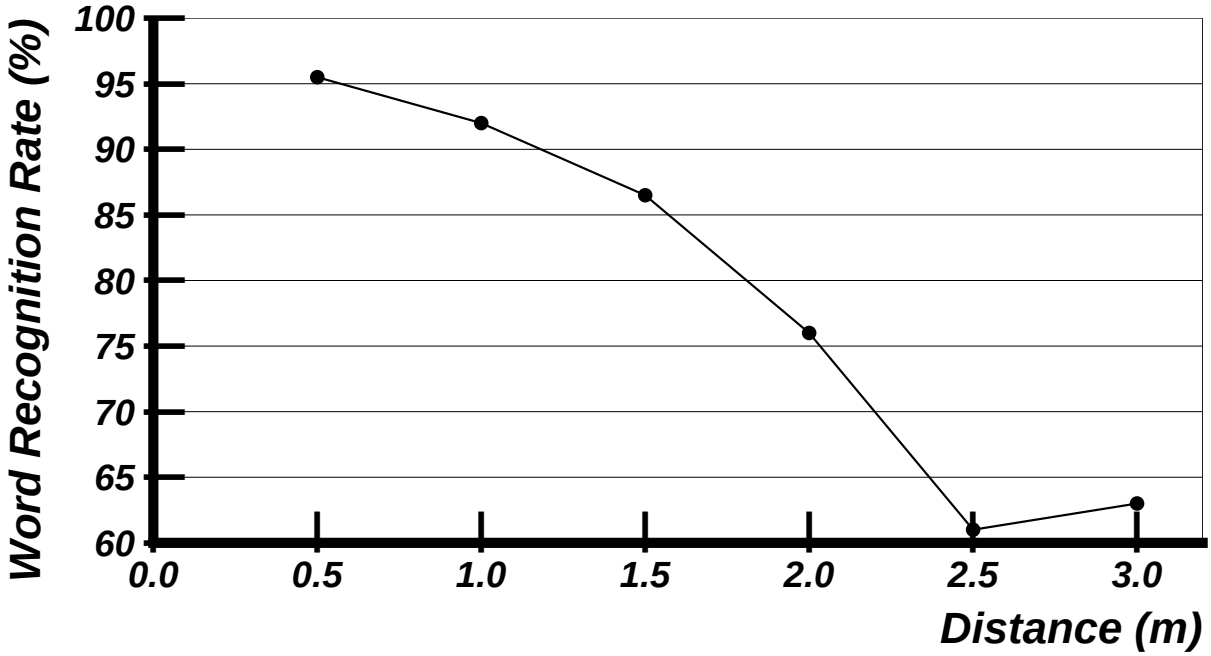


Figure 3.1: The isolated word recognition at various distances (4, Figure 1)

of the IPD and the IID based on Dempster-Shafer Theory [76]. A brief factor of each direction θ is estimated by an IPD brief factor $B_{IPD}(\theta)$ and an IID brief factor $B_{IID}(\theta)$ as follows:

$$B_{IPD+IID}(\theta) = 1 - (1 - B_{IPD}(\theta))(1 - B_{IID}(\theta)) \tag{3.1}$$

The effective distance of sound source localization on average and the standard deviations were estimated in our laboratory (Figure 3.2). Three directions were evaluated separately. The horizontal direction was specified from right (0 [degree]) to left (180 [degree]), and the center was 90 [degree].

Some robots [77][78] use the MUltiple Signal Classification (MUSIC)[79] method extended to a broadband signal with eigenvalue weighting[80][81]. We also estimated the effective distance of sound source localization based on the MUSIC method.

We denote the input vector as $\boldsymbol{x}(\omega, t) = [X_1(\omega, t), \cdots, X_M(\omega, t)]^T$, where $X_m(\omega, t)$ denotes the short-time Fourier transform of the input signal to the mth microphone.

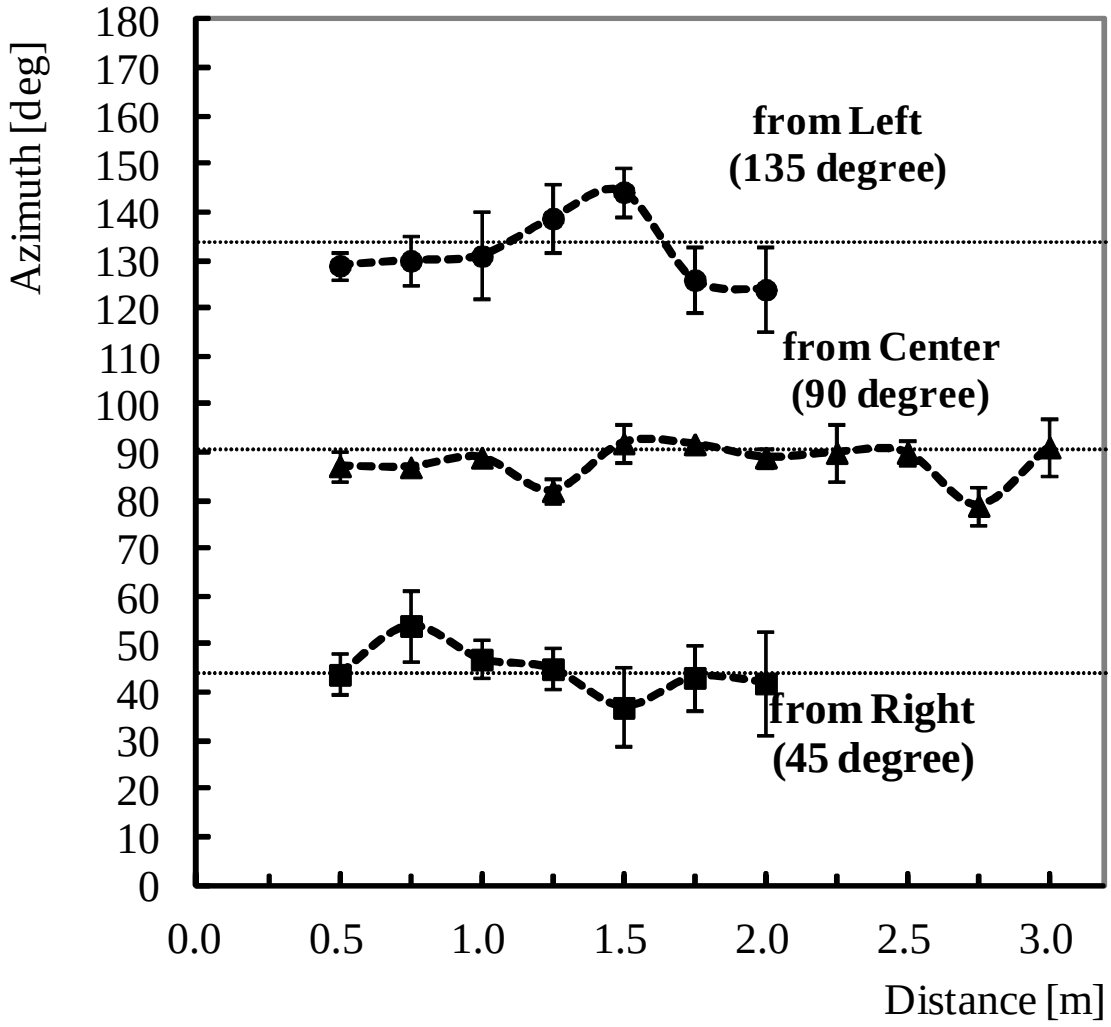


Figure 3.2: The sound source localization at various distances (IPD·IID)

From this input vector, the spatial correlation is estimated as

$$\boldsymbol{R}(\omega) = E[\boldsymbol{x}(\omega, t)\boldsymbol{x}^H(\omega, t)]. \tag{3.2}$$

Using the eigenvectors of $\boldsymbol{R}(\omega)$ corresponding to the smallest $M-N$ eigenvalues, $e_{N+1}, \cdots, e_M$, the MUSIC spatial spectrum estimator is defined as

$$P(\theta, \omega) = \frac{|\boldsymbol{a}(\theta, \omega)|^2}{\sum_{m=N+1}^{M} |e_m^H \boldsymbol{a}(\theta, \omega)|^2}, \tag{3.3}$$

where M and N denote the number of microphones and the number of sound sources, respectively. The symbol $\boldsymbol{a}(\theta, \omega)$ denotes the location vector of the virtual source in the arbitrary direction θ. The elements of the location vector are the transfer functions of the direct path from the virtual source to the microphones. To estimate the final spatial spectrum for the broadband input, Equation 3.3 is averaged over the frequency of interest as

$$\bar{P}(\theta) = \sum_{\omega=\omega_l}^{\omega_h} \bar{\lambda}(\omega) P(\theta, \omega), \tag{3.4}$$

where $\bar{\lambda}$ is the eigenvalue weight [79] defined as

$$\bar{\lambda} = \sum_{n=1}^{N} \lambda_n. \tag{3.5}$$

The symbol λ_n is the nth eigenvalue of $\boldsymbol{R}(\omega)$. The eigenvalues are assumed to be sorted in descending order. By doing this, the frequency bins in which the power of the directional signal is dominant have larger weights. The range $[\omega_l, \omega_h]$ denotes the frequency range of interest.

The effective distance of sound source localization on average and the standard deviations based on the MUSIC method were estimated in our laboratory (Figure 3.3). Three directions were evaluated separately. The horizontal direction was specified from right (0 [degree]) to left (180 [degree]), and the center was 90 [degree].

The localization errors of both methods were small for distances less than about 3 [m]. Therefore, sound source localization should be stable up to around 3 [m].

Face Localization

We use MPIsearch [82] for robust face detection. A robot can measure the distance and direction to a person based on the average size of a person's face as shown in Figure 3.4.

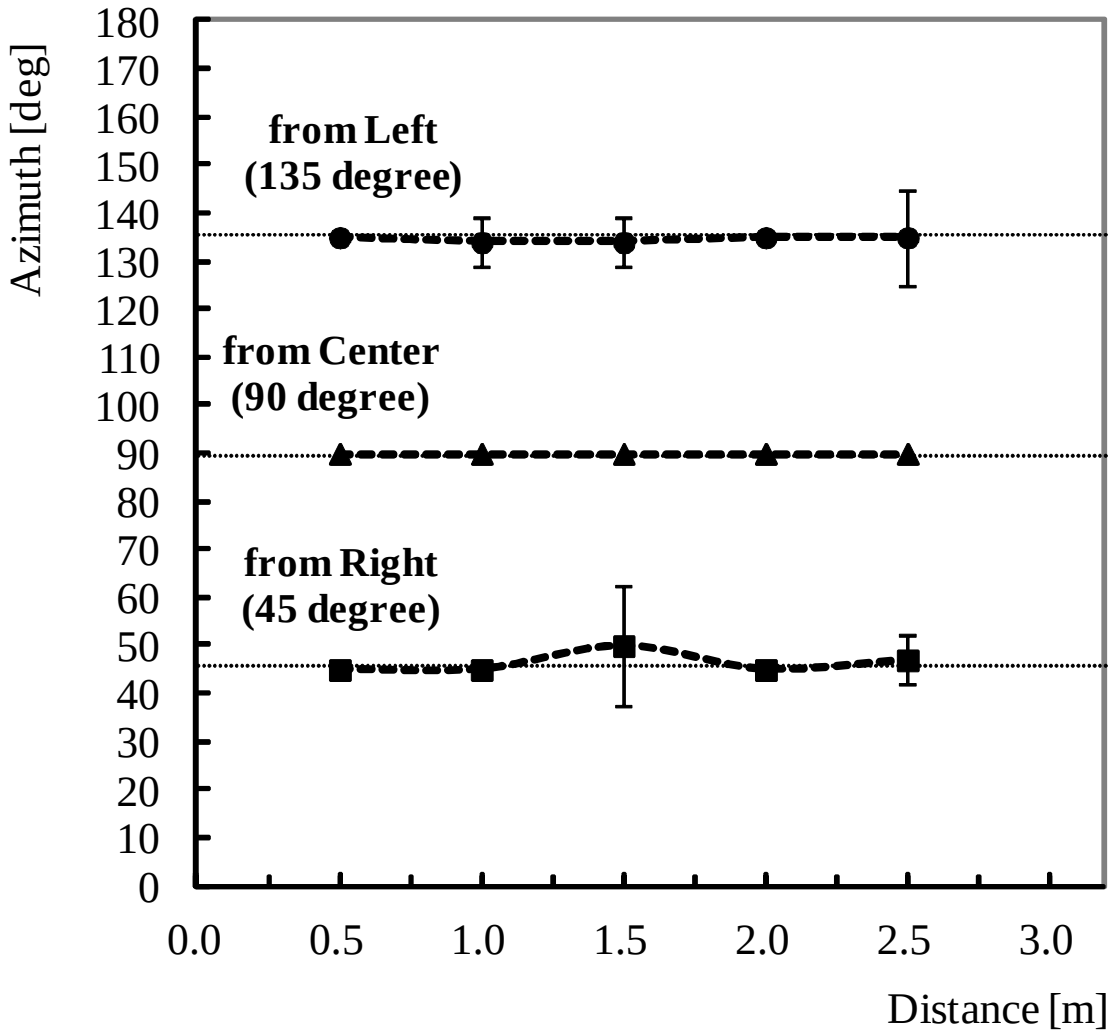


Figure 3.3: The sound source localization at various distances (MUSIC)

MPIsearch requires an image at least 12 by 12 pixels to detect a face. Such images correspond to a distance of 4 to 5 [m]. In general, the effective distance of face localization is up to the public distance.

Advantages and Disadvantages of Functions

The advantages and disadvantages of tactile recognition, sound source localization, and face localization are shown in Table 3.1. While tactile recognition can localize a person within the length of a person's arm, it cannot detect the direction to the person precisely. While sound source localization can detect the direction to the person exists and is not

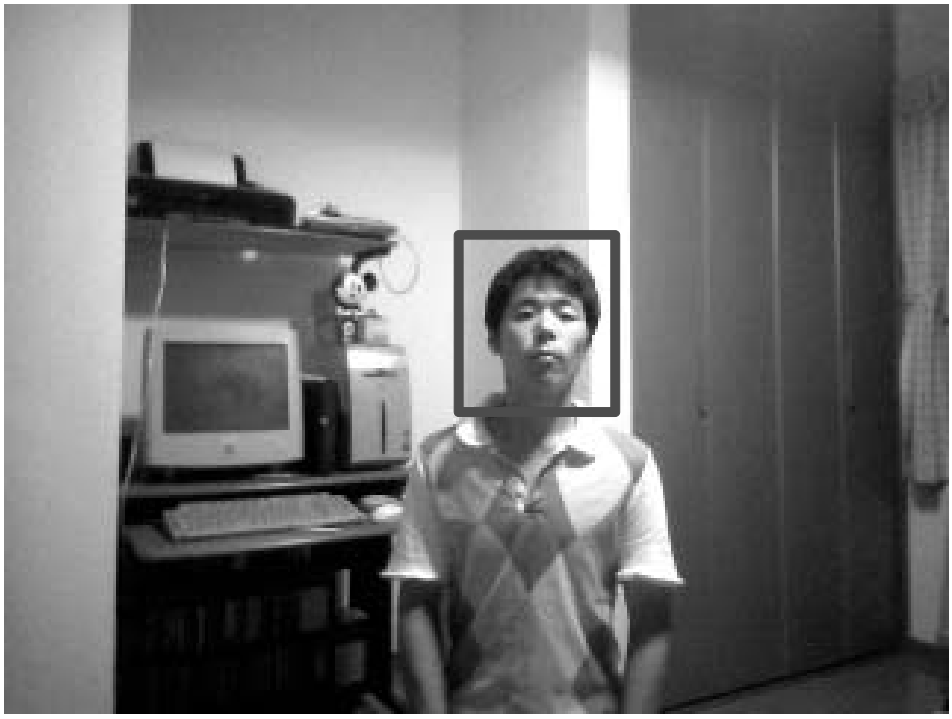


Figure 3.4: The face localization by MPIsearch

affected by occlusion, it is affected by environmental sound (noise). While face localization can detect not only the distance but also the direction to the person, it suffers if the lighting is poor.

Table 3.1: The advantages and the disadvantages of the robot functions

Function	Advantages	Disadvantages
Tactile Recognition	near distance detection high reliability	weak direction detection
Sound Source Localization	direction detection no occulusion effects	mixed sound effects cannot measure the distance
Face Localization	direction detection distance detection	light effects occulusion effects

Considering these factors, the integration of several functions into a robot enables a robot to localize people more robustly. For example, if poor lighting impairs face localization, tactile recognition and sound source localization can be used instead.

Table 3.2: The relationship between the distance and the function

	Intimate Distance	Personal Distance	Social Distance
input	Tactile Recognition Speech Recognition Face Localization Sound Localization	 Speech Recognition Face Localization Sound Localization	 Face Localization Sound Localization
output	Utterance Dialogue Motion for Near Field (Touch Reaction)	Utterance Dialogue Motion for Distant Field (Gazing)	Utterance Motion for Distant Field (Gazing)

3.1.3 Interaction Distance and Effective Distance of Functions

The relationship between the interaction distance and the effective distance for the three functions is shown in Table 3.2. As shown in the Table 3.2, effective distance for the functions can correspond to the interaction distance effectively.

3.2 Friendliness Space Map

3.2.1 Design of Friendliness Space Map

In various environments, the sensor inputs capture noise. Moreover, the sensor functions a robot can use effectively differ depending on the distance between the robot and each person.

In other relational studies, the robot always used all sensors and interacted with people by focusing on the people. In our study, the robot interacted with people by focusing on the "space" of the people. In particular, the robot acted based on the space around the robot, segmented as described in Table 3.2.

Given the size of a person's face and the accuracy of the robot's functions, the direction element of space must be segmented to some extent. We segmented the space every 15 degrees based on the average size of the human face (16 [cm] $\times$ 23 [cm]) and the errors of functions within the personal distance.

To identify the intimate space for the robot to interact with, we defined polar coordinates as shown in Figure 3.5. These coordinates, which are segmented into cells, are called a "Friendliness Space Map". Our robot calculates the "friendliness" of a cell (r, θ) using information about the location of people and comfortable/uncomfortable stimuli.

To calculate the friendliness, when a function is initiated by sensor input, our robot calculates the Human Existence Degree (HED), which shows whether people exist or not, of cells within the effective area of each function. For example, three areas where our robot calculated the HED are shown in Figure 3.5: (1) in the case the right side of the robot is touched, (2) in the case the robot detects sound, (3) in the case the robot detects face.

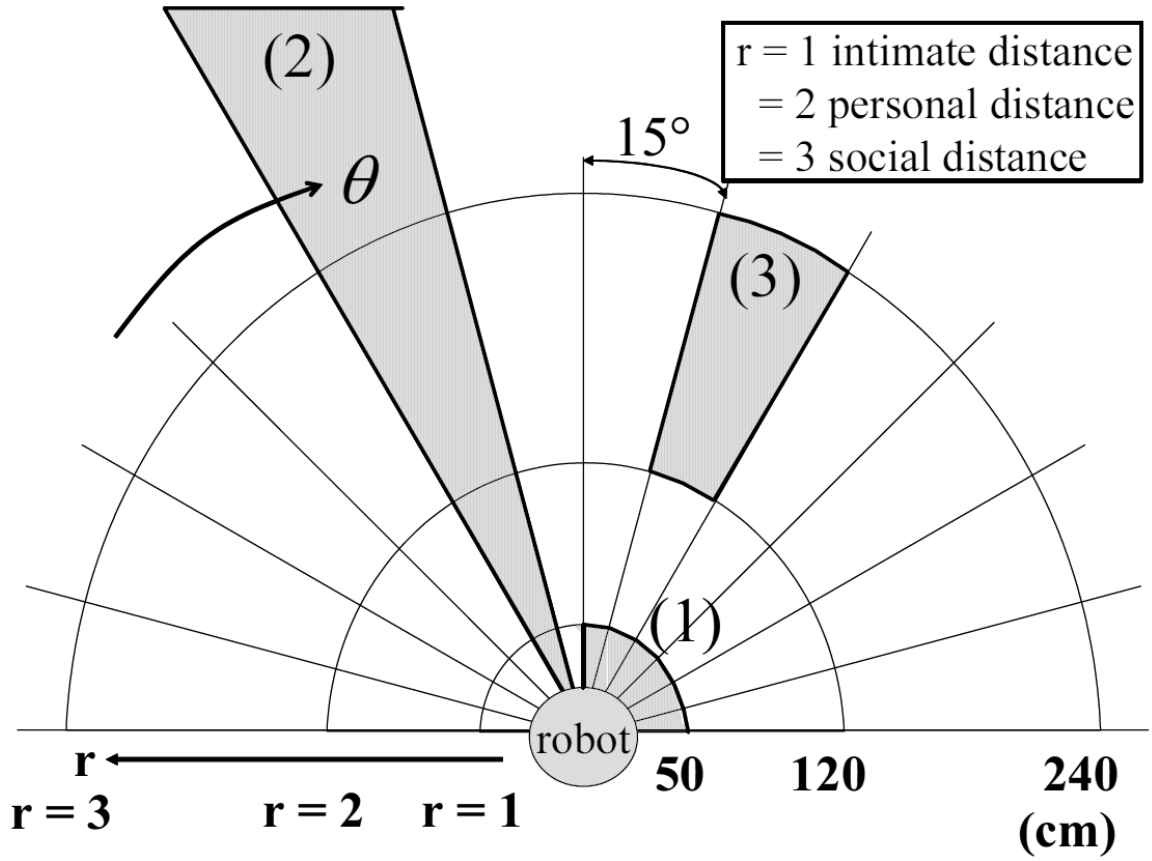


Figure 3.5: The friendliness space map and the effective area of functions (3, Figure 1)

The effects of interaction using this map are as follows.

- Since a robot can change its motion and select an interaction partner based on the friendliness of various spaces, it can interact with multiple people simultaneously in various environments.

- The action selection based on space can also be applied to various other objects.

3.2.2 Definition of Human Existence Degree by Integration of Functions

In each cell on the map, the HED is calculated by taking advantage of the integrated functions. When a function k locates a person at time t_{k0}, it calculates the HED, $L_{k,t,r,\theta}$ of cell (r, θ) within the effective function area at time t, as shown Equation 3.6. The $k(k = 1, 2, 3)$ is the functions, d_k is the damping ratio (which is decided based on the degree of confidence obtained by previous experiments of each function) , and t_{k0} is renewed every time function k operates.

$$L_{k,t,r,\theta} = \exp[-d_k(t - t_{k0}] \tag{3.6}$$

The HED calculated by integration of all functions, $E_{t,r,\theta}$ of cell (r, θ) at time t is defined as the sum of the HED of each function:

$$E_{t,r,\theta} = \sum_{k=1}^{3} L_{k,t,r,\theta} \tag{3.7}$$

3.2.3 Shift in Friendliness by Stimulus

The cells on the Friendliness Space Map are affected by the kind of stimulus. Our robot recognizes two kinds of stimuli by using tactile recognition. One is uncomfortable stimuli, such as hitting the robot's head or touching the robot's bust. The other is comfortable stimuli, such as patting the robot's head. Since tactile recognition cannot localize people precisely, we assume the person delivering the stimulus is in the cell with the highest human existence degree within the intimate distance. That is, it is cell $(1, \theta)$, as obtained using

$$\theta = \mathrm{argmax}_{\theta} E_{t,1,\theta} \tag{3.8}$$

If the stimulus occurs at time t_{C0}, we define the Comfortable Degree (CD), $C_{t,1,\theta}$ of cell $(1, \theta)$ selected at time t as shown in Equation 3.9, where d_C is the damping ratio, v is the kind of stimulus (v 1 is a comfortable stimulus and v -1 is an uncomfortable stimulus), and t_{C0} is renewed every time a stimulus is received.

$$C_{t,r,\theta} = v \times \exp[-d_C(t - t_{C0})] \tag{3.9}$$

3.2.4 Definition of Friendliness

The Friendliness Space Map is renewed and consists of both the HED and the CD obtained using the robot's functions. The friendliness, $I_{t,r,\theta}$ of cell (r, θ) at time t is defined as the sum of the HED and the CD as shown in Equation 3.10, where W_L and W_C correspond to the weights of the HED and the CD, respectively. In this time, we make W_C bigger than W_L because we want a robot to be sensitive to the stimulus.

$$I_{t,r,\theta} = W_L \times E_{t,r,\theta} + W_C \times C_{t,r,\theta} \tag{3.10}$$

3.3 Summary

In this chapter, we describe how to localize and select an interaction partner from among multiple people. We use the interaction distance which shows friendliness between a person and the others. In the psychology theory, the interaction distance can be classified into roughly four groups: intimate, personal, social, and public. Moreover, the effective distance of robot's functions corresponds to the interaction distance. Therefore, we divide the space around the robot based on the interaction distance. Mapping the friendliness which is calculated by the multiple sensors' input onto each divided space, the robot interacts with the highest friendliness spaces selectively. We call the divided space "friendliness space map".

Chapter 4

Implementation and Evaluation of Detecting People for Interaction Robot

This chapter introduces our interaction robot SIG2 about both hardwares and softwares. This chapter also evaluates the method detecting the interaction partner.

4.1 Design of Interaction Based on the Friendliness Space Map

4.1.1 Interaction Robot SIG2

Hardware of SIG2

The platform we used is the humanoid robot SIG2 shown in Figure 4.1. SIG2 is a humanoid robot whose body is fixed on the vehicle (ActivMedia Robotics Pioneer2-AT). The body is made of FRP.

It has 19 tactile sensors on its head and upper body, a microphone ("ear") on each side of its head, and two cameras ("eyes") in its head. To reduce a sound noise, each microphone is embedded at the eardrum of a human outer ear model made of silicon, as shown in Figure 4.1. We use the microphone (ME102) made by SENNHEISER. We use the camera (QN42H) made by Elmo. The tactile sensors consist of touch sensors and sillicon [83]. The position of 19 tactile sensors shows Table 4.1. The touch sensors are sheet-like and made of polyvinylidene fluoride (PVDF) produced by MSI·Tokyo sensor.

SIG2 has hardwares for outputs , speakers and motors. SIG2 utters and gestures by using a speaker and three motors in its head. The neck of SIG2 is consist of 3 motors. They enable SIG2 to take a nod and cock its head. The motor driver is TITech Driver

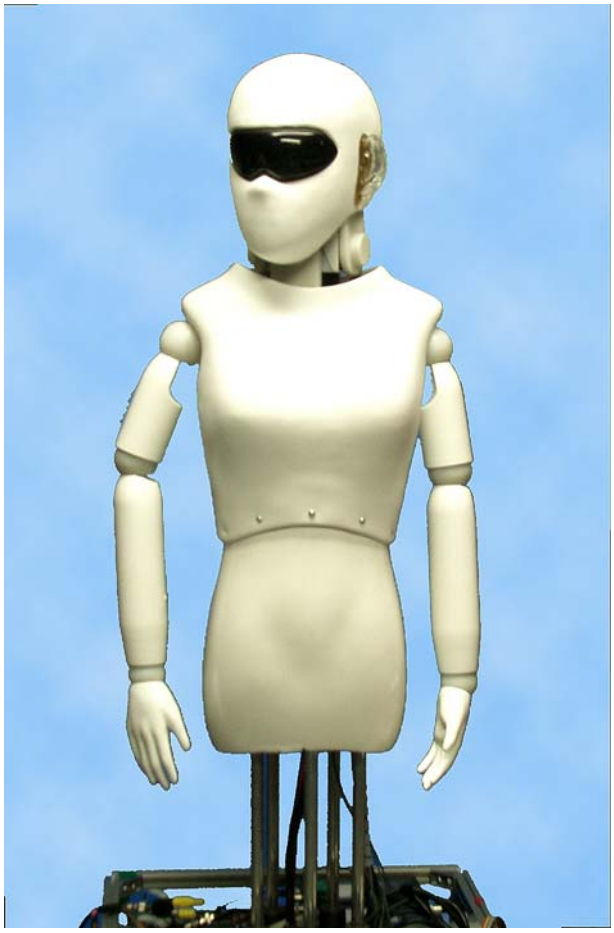


Figure 4.1: Our robot SIG2 (4, Figure 4)

PC-0121-2.

Software of SIG2

SIG2 is equipped with tactile recognition, speech recognition, sound source localization (the IPD·IID method), and face localizationas (MPIsearch) as shown in Chapter 3. The tactile recognition recognizes the spot on the robot touched by a person based on the positions of the sensors as shown in Table 4.1. It recognizes two kinds of contact (hitting and patting) using the touch duration. Figure 4.2 shows an example of the sensor data. The speech recognition recognizes the numbers from 1 to 15, and "yes", and "no".

The output functions are a tactile reaction function, a game dialogue function, an interaction partner selection function, a trace face function and a trace sound function.

The tactile reaction function enables SIG2 to perform five types of actions such as a delighted action or a sad action based on both the spot touched and the kind of stimulus.

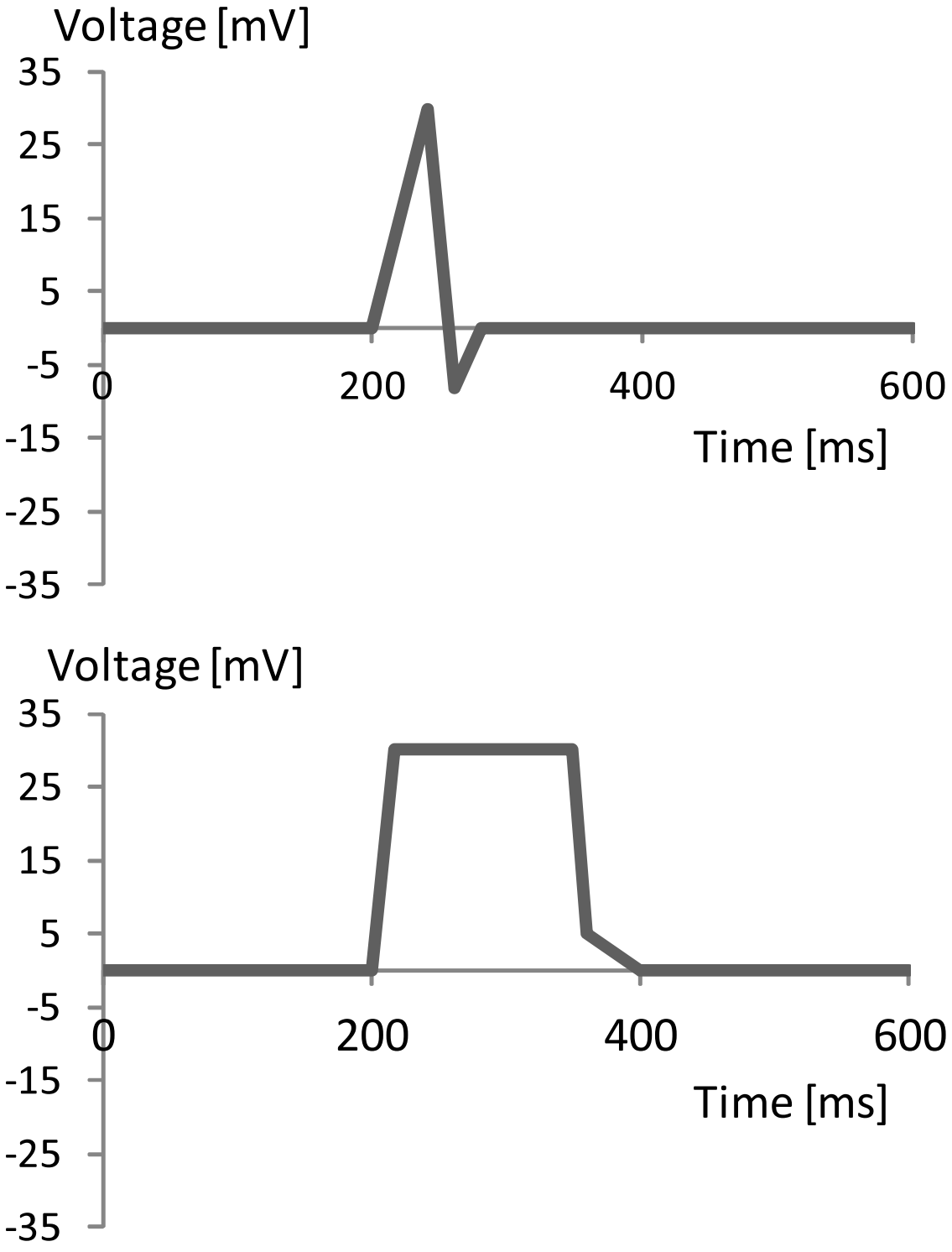
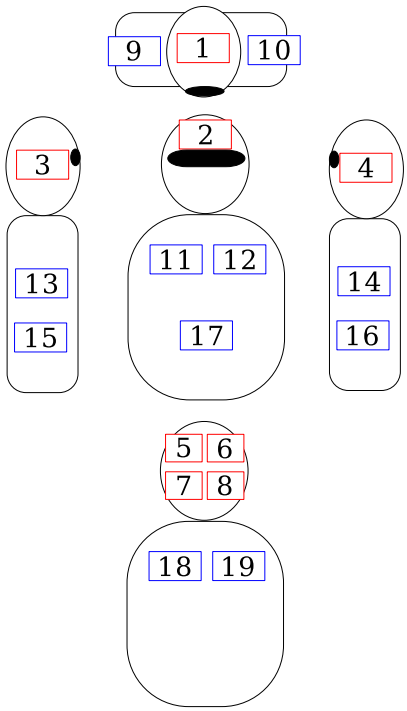


Figure 4.2: The output of tactile sensors (upper: hit, lower: pat)

Table 4.1: The position of the tactile sensors on SIG2

No.	Position	No.	Position
1	top of head	9	right shoulder
2	forehead	10	left shoulder
3	right of head	11	upper right chest
4	left of head	12	upper left chest
5	upper left occipital	13	right armpit
6	upper right occipital	14	left armpit
7	lower left occipital	15	right flank
8	lower right occipital	16	left flank
		17	abdominal part
		18	left back
		19	right back



The reaction after the tactile recognition is shown in Table 4.2.

Table 4.2: The reaction after the tactile recognition

Position	head	shoulder	chest
hit	speak “ouch” nod no	speak “what?” turn to the person	speak “no” look away
pat	speak “thank you” cock its head	speak “what?” turn to the person	speak “no” look away

The game dialogue function enables SIG2 to play a game using speech recognition if there is an intimate person within the personal distance by using speech recognition. In this game, SIG2 and its interaction partner say random numbers from 1 to 15 to each other. They can repeat the number up to four times at once. The first one who says a number that has already been said loses. SIG2 uses gestures and utterances that match the situation of the game. Figure 4.3 shows the flow of the game dialogue.

The intimate person selection function makes SIG2 turn on the cell direction that has

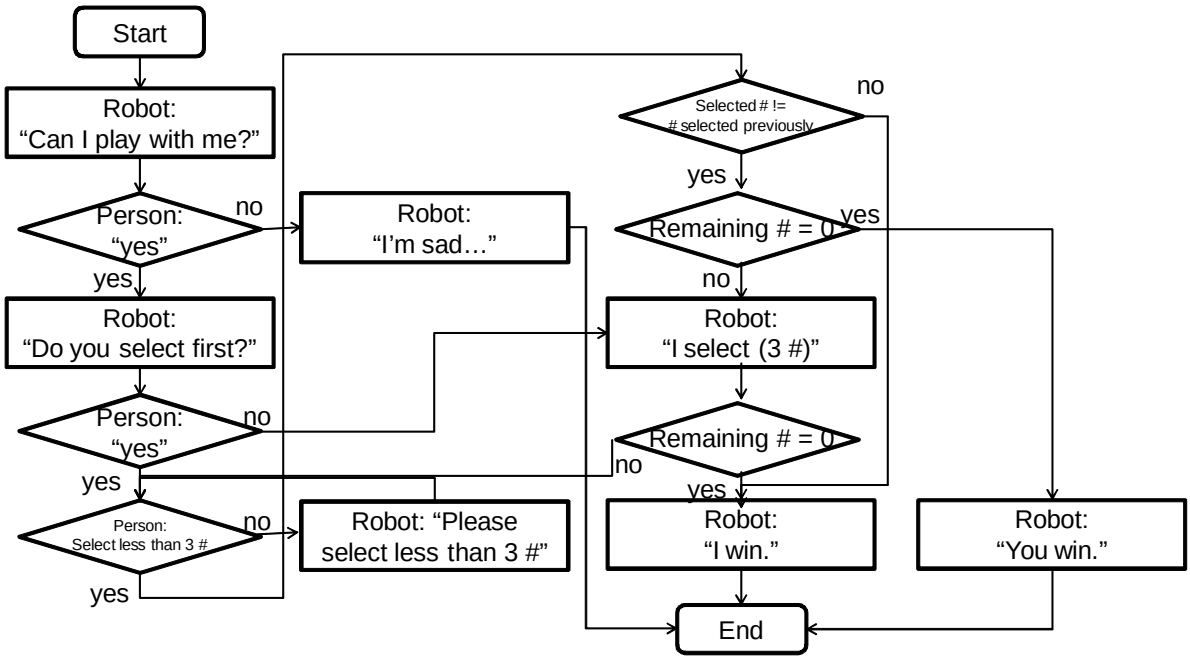


Figure 4.3: The flow of the game dialogue

the highest friendliness level within the personal distance, after gesturing and uttering using other output functions.

The face trace function and the sound trace function enable SIG2 to gaze at the direction where it finds a person's face or hears a sound.

4.1.2 Motion of SIG2 during Interaction

More specifically, SIG2 interacts with people as follows.

1. SIG2 turns on the direction calculated by tactile recognition, face localization, or sound source localization. If SIG2 uses tactile recognition, it acts based on both the kind of stimulus and the spot touched in accordance with the outputs of the tactile recognition.

2. After referring to the friendliness space map, SIG2 renews it based on the results of person localization and stimulus type.

3. If the stimulus is comfortable and the friendliness of the cell within the personal distance exceeds a threshold, SIG2 plays a game with the person in that cell.

4. SIG2 turns on the direction of the cell that has the highest friendliness level on the friendliness space map.

Figure 4.4 shows the whole system implemented into SIG2. Considering the accuracy of modules discussed in Section 3.1.2, the damping ratios of the tactile recogntion d_1, the sound source localization d_2, and the face localization d_3 used by Equation 3.6 are 1.0, 1.5, and 2.0 ($d_1 < d_2 < d_3$) , respectively. In order to react susceptibly during short interaction, the damping ratio d_C of comfortable degree used in Equation 3.9 is 4.0, which is larger than ($d_1 < d_2 < d_3$). Moreover, The weights of the human existance degree (W_L) and the comfortable degree (W_C) used in Equation 3.10 are 1.0 and 4.0, respectively.

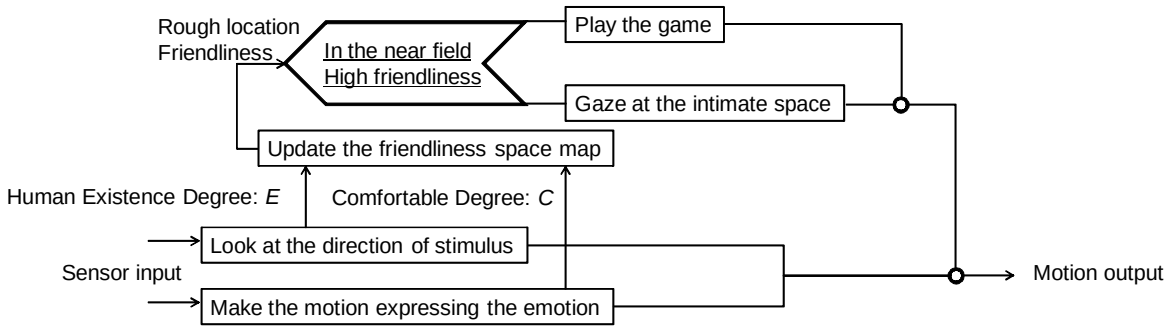


Figure 4.4: The system implemented into SIG2

4.2 Effectiveness Evaluation of the Person Localization by SIG2

4.2.1 Aim and Sequence of Effectiveness Evaluation at Event Hall

To determine whether a person is in the direction where SIG2 feels intimate in an actual environment, we compared the accuracy of sound source localization, which is the most accurate of the three functions, with the accuracy of proposed method, at an event hall of a big facility assumed a big shopping store. Testing was done during the daytime, so the event hall was illuminated by both natural and artificial light. Moreover, a hall announcements were broadcasted regularly. Testing was done using seven pairs of participants. Experimental steps are as follows:

1. We explained to the participants the input functions of SIG2.

2. SIG2 interacted with each pair for about 5 [min]. The evaluation criteria were the recall ratio (R), precision ratio (P), and F measure as shown in Equation 4.1, 4.2 and 4.3. T [sec] denotes a duration when the detected direction corresponded with one of the people. T_r [sec] denotes a duration of their interaction. T_p denotes a duration when the system detected people.

$$R = \frac{T}{T_r} \tag{4.1}$$

$$P = \frac{T}{T_p} \tag{4.2}$$

$$F = \frac{2PR}{P+R} \tag{4.3}$$

We also investigated whether our method detects interaction partner while the robot stands by and interacts with people. Two labelers observe their interaction and select interaction partners whom our robot should interact with. We evaluate our method by two values E_1 and E_2 as shown in Equation 4.4 and 4.5:

$$E_1 = \frac{T_{rob}}{T_{lab}} \tag{4.4}$$

$$E_2 = \frac{T_{exist}}{T_{out}} \tag{4.5}$$

Here, T_{lab} shows the duration when two labelers select same partners. T_{rob} shows the duration when two labelers and our robot select same partners. T_{out} shows the duration when our robot outputs detecting people. T_{exist} shows the duration when our robot outputs detecting people correctly.

4.2.2 Results

The relationship between the distribution of cells which had the highest friendliness level at the intimate distance and the directions in which there were people is shown in Figure 4.5. Two people interacted with SIG2 at directions of around 60 [deg] and 105 [deg] which correspond to a person on left and a person on right in Figure 4.5 respectively. In Figure 4.5 , we can see that there were people in the cell with the highest friendliness level. The gray value in Figure 4.5 shows the friendliness level. The black denotes the highest level.

The recall ratio, precision ratio, and F measure are shown in Table 4.3. The F measure with the developed method was higher than with only sound source localization.

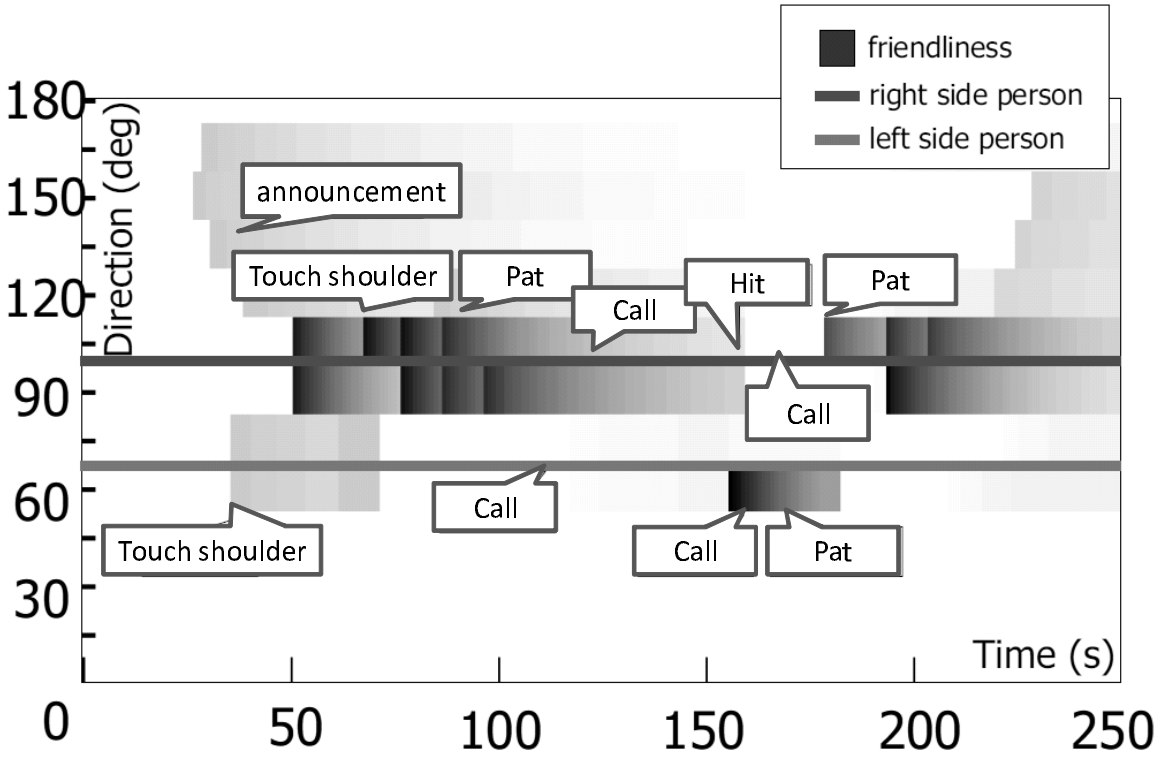


Figure 4.5: The relationship between the friendliness distribution and the location of people (3, Figure 6)

Table 4.3: The accuracy of the person localization

	Only Sound Localization	Friendliness Space Map
Recall	0.33	0.71
Precision	0.52	0.83
F measure	0.40	0.76

The experimental results also show that E_1 denotes 0.95 and E_2 denotes 0.87. We think that E_1 is high enough to detect people who call robots. E_1 is higher than E_2, which shows that our robot can especially select people whom humans (labelers) can select by only observing the interaction.

4.3 Impression Evaluation of Interaction Using Friendliness Space Map by SIG2

4.3.1 Aim and Sequence of Impression Evaluation

We investigated whether our method enabled SIG2 to make a plausible and friendly impression when interacting with several people simultaneously. We asked 27 visitors (men and women ranging in age from 20 to 54) to interact with SIG2 at the event hall and then fill out a questionnaire. The experimental conditions were the same as described in Section 4.2 . The experimental setup is shown in Figure 4.6.

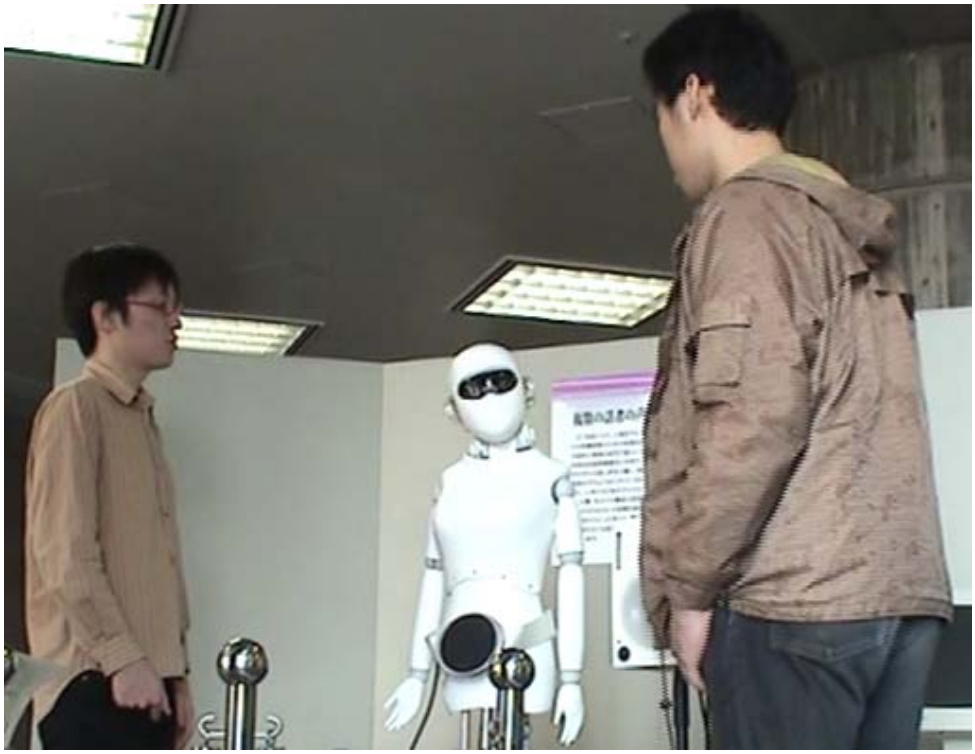


Figure 4.6: The experimental setup (3, Figure 4)

Each groups of visitors interacted with SIG2 two times, and SIG2 used a different behavior each time. One time it behaved based on the friendliness space map, as described in Chapter 3. The other time it did not use friendliness space map to isolate the effects of our method. In the latter, SIG2 turned in the direction calculated by three functions and played the game regardless of the friendliness level if someone was within the personal distance. SIG2 selected which behavior to use at the beginning randomly. Each group interacted with SIG2 for about 5 [min] each time. Then, they filled in a

questionnaire, rating 28 adjective pairs (in Japanese) on 1-to-7 scales, where 7 means the positive adjectives fit very well (adjectives in the leftmost column in Table 4.4), based on the SD method. This evaluation method is based on "Psychological analysis on human-robot interaction"[84]. The deviation in the case of using 1-to-7 scales is larger than the deviation in the case of using 1-to-3 scales or 1-to-5 scales. Therefore, it is suitable for comparison experiments to use 1-to-7 scales.

4.3.2 Results

A factor analysis was conducted on the SD method ratings for the 28 adjective pairs using the results from 54 (27×2) questionnaires. We used four factors based on the results obtained by [84], which concluded that four factors are appropriate for the interaction. Moreover, the cumulative proportion was 59.4 %, which is similar to the cumulative proportion (57.8 %) obtained by [84]. The factor matrix along with the factor loadings, are listed in Table 4.4.

Referring to the adjective pairs that have loadings greater than 0.6, the first factor contains "Distinct", "Exciting", and so on, and the second factor contains "Kind", "Friendly", and so on. The first and second factors are similar to the ones obtained by Kanda et al. [84]. Therefore, we think the two types of SIG2 behaviors can be compared meaningfully using the first and second factors. In particular, one factor obtained by [84] was named the "enjoyment factor" because the adjectives in that factor were related to the level of enjoyment. The top three adjectives with loadings in the "enjoyment factor" were "Exciting", "Pleasant", and "Likable". Moreover, one factor obtained by [84] was named the "familiarity factor" because the adjectives in this factor were related to the familiarity. The top three adjectives with loadings in the "enjoyment factor" were "Kind", "Favorable", and "Friendly". In our work, the top three adjectives with loadings in the first factor were "Exciting", "Pleasant", and "Likable", too. The top three adjectives with loadings in the second factor were "Kind", "Favorable", and "Friendly", too. Since adjectives with loadings in these factors are the same as the adjectives extracted by Kanda [84], our first and second factors were named the enjoyment factor and familiarity factor, respectively. We think the enjoyment factor is based on the enjoyment of the interaction itself. We also think the familiarity factor is based on the relationship between the robot and people. These are the main differences between the two factors.

Table 4.4 shows the average and standard deviations of the impression scores for the

Table 4.4: The adjective paris and the result of the factor analysis

Adjective pairs		Factor 1	Factor 2	Factor 3	Factor 4
Kind	Cruel	-0.103	**0.720**	0.149	-0.110
Favorable	Unfavorable	0.094	**0.689**	-0.110	-0.186
Friendly	Unfriendly	0.315	**0.681**	-0.061	-0.028
Safe	Dangerous	0.204	0.517	-0.191	0.257
Warm	Cold	0.275	0.550	0.262	-0.043
Prety	Ugly	0.220	**0.661**	0.195	0.011
Frank	Rigid	0.535	0.291	0.143	-0.004
Distinct	Vague	**0.636**	-0.099	-0.092	-0.474
Accessible	Inaccesible	0.522	0.265	0.061	-0.072
Light	Dark	0.470	0.318	0.329	0.051
Altruistic	Selfish	0.260	0.155	0.188	-0.493
Humanlike	Mechanical	0.413	0.289	0.186	-0.035
Full	Empty	**0.604**	-0.027	0.058	0.119
Exciting	Dull	**0.857**	0.002	-0.196	0.023
Pleasant	Unpleasant	**0.805**	0.138	-0.159	0.101
Likable	Dislieable	**0.857**	0.122	-0.034	-0.137
Interesting	Boring	0.497	0.442	-0.246	0.027
Good	Bad	**0.734**	0.151	-0.183	0.004
Complex	Simple	0.045	0.058	0.419	0.139
Rapid	Slow	-0.103	0.007	**0.910**	-0.153
Quick	Slow	-0.147	0.017	**0.808**	-0.109
Agitated	Calm	-0.020	-0.499	0.484	0.109
Active	Passive	0.105	0.108	0.493	0.498
Brave	Cowardly	0.076	-0.136	0.076	**0.761**
Showy	Quiet	**0.674**	-0.448	0.378	0.110
Cheerful	Lonely	0.401	0.311	0.350	0.135
Sharp	Blunt	0.009	0.149	0.552	0.058
Intelligent	Unintelligent	**0.801**	-0.010	-0.014	-0.244

two types of behaviors for the first and second factors. Figure 4.7 shows a bar graph of the impression scores. About the first factor, t-test showed that the difference between the two types was significant at the 0.01 level ($p = 0.004$, $t(13) = 3.45$, $d = 0.92$, $r = 0.69$). Here, d and r denote the Cohen 's d and the pearson product-moment correlation coefficient, respectively. About the second factor, t-test showed that the difference between the two types was significant at the 0.05 level ($p = 0.02$, $t(6) = 3.0$, $d = 1.16$, $r = 0.78$). The effect sizes of the first and second factors were 0.92 and 1.16, respectively, which indicates that the behavior based on the friendliness space map was considered to have more positive adjectives.

Table 4.5: The comparison of the impression scores

	First Factor		Second Factor	
Type	Average	S. D.	Average	S. D.
Based on Map	4.59	1.58	4.79	0.97
Ignore Map	4.35	1.75	4.65	0.85

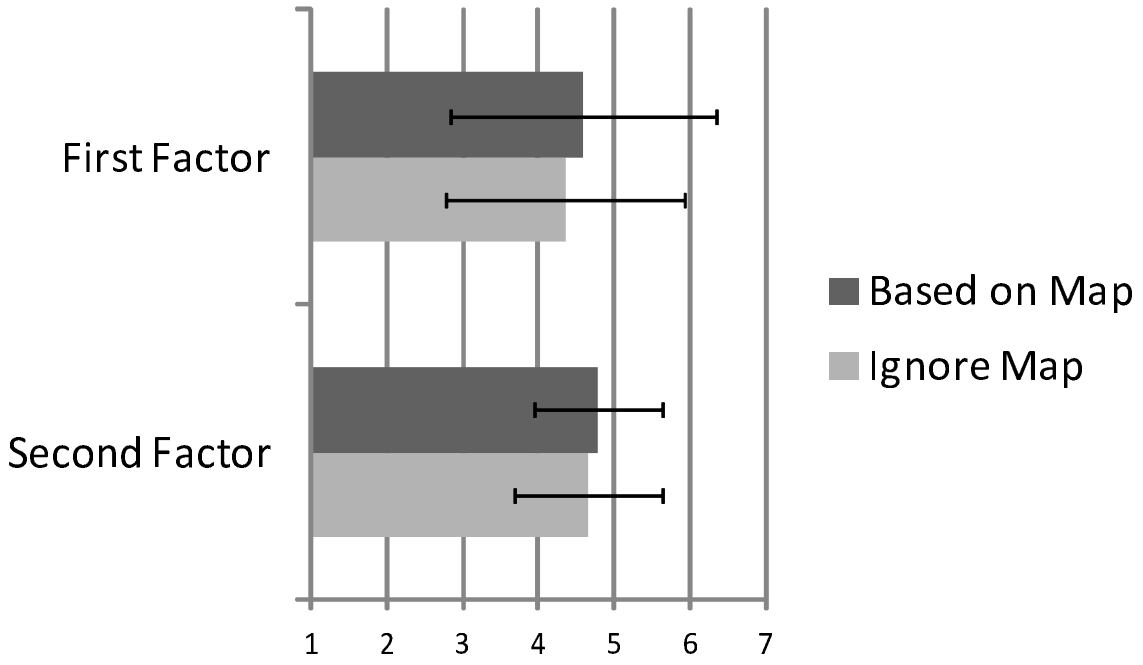


Figure 4.7: Comparison of Impression Scores of Using Friendliness Level (3, Figure 7)

4.4 Summary

In this chapter, we evaluate our interaction partner selection method by using humanoid robot SIG2. The evaluation aims to confirm that our method can localize people and that SIG2 can interact friendly with multiple people by using our method. SIG2 has four functions to calculate friendliness and to localize people. The four functions are tactile recognition, speech recognition, sound localization and face localization. Integrating four functions on the friendliness space map can localize people more accurately. Moreover, results of questionnaires to the interaction between SIG2 and multiple people show that SIG2 can interact more friendly with multiple people by using friendliness based on the distance.

Chapter 5

Obstacle Classification and Location by Using a Mobile Omnidirectional Camera Based on Tracked Floor Boundary Points

This chapter provides how to find the floor boundary points and describes the obstacle classification method based on the result of tracking floor boundary points.

5.1 Floor Boundary Points Detection

5.1.1 Floor Detection by Ward's Clustering

We use floor colors for floor detection because floor colors are generally simple. Previous works use the Gaussian Mixture Model (GMM) for specific color detection [85]. The GMM can detect many specific colors, increasing a number of a mixed Gaussian. However, we have to evaluate the GMM many times in order to decide parameters such as the number of mixed Gaussian. Therefore, it is difficult for robots to apply the GMM to various environments quickly and accurately just after they start up.

Our robot learns representative colors of the floor by itself based on the distribution of floor color data without prior setting. Considering the distribution, our floor detection method can adjust more easily than the GMM can and detects the floor as accurately as the GMM does. Here, in order to detect the representative colors of the floor, we use Ward's clustering [86], which is one of the hierarchical clustering methods. Our robot selects the representative colors by Ward's clustering as follows.

1. Our robot takes an image and gets N color data from pixels to which the close area

around it is projected. In an initial state, each datum shows a representative color. A cluster of color data that are similar to the representative color i is denoted by C_i.

2. We choose two clusters C_1 and C_2 that minimize D as shown in Equation 5.1 and Equation 5.2, and create a new claster C_k that consists of the data in both C_1 and C_2. Let c_i denote an average color vector in the cluster C_i.

$$D(C_1, C_2) = d(C_1 \cup C_2) - d(C_1) - d(C_2) \tag{5.1}$$

$$d(C_i) = \sum_{x \in C_i} \|x - c_i\| \tag{5.2}$$

3. In step 2, when C_k satisfies both Equation 5.3 and Equation 5.4, it is decided that c_k is the representative color and data in C_k are not used for following loops. When C_k satisfies only Equation 5.3, data in C_k are just not used for following loops. T_D and T_N is a constant threshold, $|C_k|$ is a number of the data in C_k.

$$\min_{k \neq i} D(C_k, C_i) > T_D \tag{5.3}$$

$$|C_k| > T_N \tag{5.4}$$

4. Step 2 and 3 continue until all data are not used.

Because Ward's clustering considers the distribution of data, each cluster is identified easily by Mahalanobis distance. A color datum I is classified as floor color when we find a C_m that satisfies Equation 5.5. μ_m, Σ_m and σ denote an average vector, a covariance matrix of data in C_m and a threshold, respectively.

$$\sqrt{(I - \mu_m)^T \Sigma_m^{-1} (I - \mu_m)} < \sigma \tag{5.5}$$

When a robot uses an omnidirectoinal camera mounted on its head, the floor is projected to around the image center. Therefore, our robot classifies the pixels from center to outer by applying Equation 5.5 . If our robot finds continuous p pixels that do not satisfy Equation 5.5, a floor boundary point is detected at the position where the first pixel in p pixels is located.

5.1.2 Transforming Coordinates of Floor Boundary Points from Image Coordinates to Robot Coordinates

In the case of using an omnidirectional camera incorporating a hyperbolic mirror, a position (X, Y, Z) on the robot coordinates is projected to a position (x, y) on the image coordinates as follows. Constant b and c denote proper parameters of the mirror, and f denotes a focal distance. Figure 5.2 shows the relationship between (X, Y, Z) and (x, y).

$$x = \frac{Xf(b^2 - c^2)}{(b^2 + c^2)(Z - c) - 2bc\sqrt{X^2 + Y^2 + (Z - c)^2}} \tag{5.6}$$

$$y = \frac{Yf(b^2 - c^2)}{(b^2 + c^2)(Z - c) - 2bc\sqrt{X^2 + Y^2 + (Z - c)^2}} \tag{5.7}$$

Many robots are equipped with an omnidirectional camera, and they can measure or know the distance from the floor to the camera while they are moving [87]. Therefore, with regard to floor boundary points, the variable Z in Equation 5.6 and Equation 5.7 become constant, and we can measure the distance from the robot to floor boundary points by applying Equation 5.6 and Equation 5.7.

In order to decide the parameters Z, b, c and f, we have drawn cross-stripes on the floor as shown in Figure 5.2. Figure 5.2 is used for determining te parameters as an input. n pairs of (X_m, Y_m) and (x_m, y_m) are acquired from the image to which n cross-points are projected. Here, (X_m, Y_m) and (x_m, y_m) denote the position of the cross-point m on the robot coordinates and the image coordinates, respectively. Using n pairs, parameters that minimize the evaluation function F as shown in Equation 5.8 are decided by the downhill simplex method [88].

$$\begin{aligned} F = \quad & \sum_{m=0}^{n-1} |x_m - X_m f \frac{b^2 - c^2}{(b^2+c^2)(Z-c) - 2bc\sqrt{X_m^2 + Y_m^2 + (Z-c)^2}}| \\ & + \sum_{m=0}^{n-1} |y_m - Y_m f \frac{b^2 - c^2}{(b^2+c^2)(Z-c) - 2bc\sqrt{X_m^2 + Y_m^2 + (Z-c)^2}}| \end{aligned} \tag{5.8}$$

For confirmation of parameters, a bird's-eye image is created by using the decided parameters. Figure 5.3 shows the bird's-eye image. The lines that make cross-stripes on the floor are not distorted, because the decided parameters are corrected. Here, 1 pixel in this bird's-eye image denotes about 5 [cm] in the real world. In the bird's-eye image, the distance between any two points on the floor is linear to the distance in the real space.

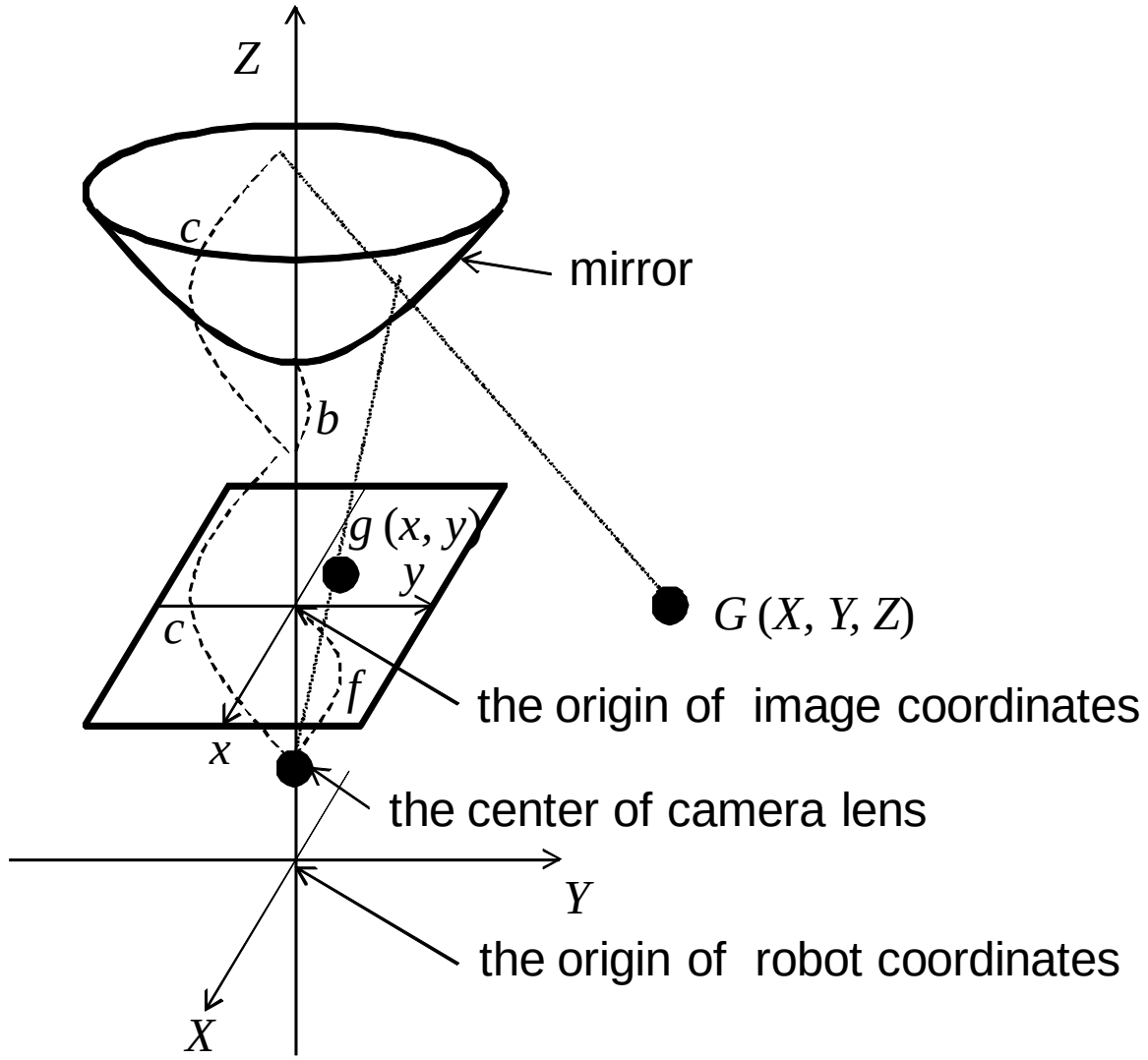


Figure 5.1: The image coordinates and the robot coordinates (5, Figure 2)

5.2 Obstacle Classification by Floor Boundary Points

5.2.1 Classification Equation

A floor boundary point m on the image at time $t-dt$ is detected by the method as shown in Section 5.1. dt depends on a processing speed. If the point m can be tracked from $t-dt$ to t correctly, the position of m at t is located correctly on the image at t. It is easy to transform the coordinates of m at $t-dt$ and t from the image coordinates to the robot coordinates $(X_m, Y_m)^{(t-dt)}$ and $(X_m, Y_m)^{(t)}$ by referring to the bird's-eye image. The relative position $(dX, dY, d\Theta)$ from $t-dt$ to t is estimated by odometry data. $\mathrm{d}\Theta$

Figure 5.2: The omnidirectional image of the cross-stripes on the floor (5, Figure 1)

is based on the direction from the center of the robot to the front of the robot at $t - dt$. When m is located at the boundary between a stable obstacle and the floor, $(X_m, Y_m)^{(t)}$ is calculated by $(dX, dY, d\Theta)$ and $(X_m, Y_m)^{(t-dt)}$, as shown in Equation 5.9.

$$\begin{pmatrix} X_m \\ Y_m \end{pmatrix}^{(t)} = \begin{pmatrix} \cos d\Theta & -\sin d\Theta \\ \sin d\Theta & \cos d\Theta \end{pmatrix} \begin{pmatrix} X_m \\ Y_m \end{pmatrix}^{(t-dt)} + \begin{pmatrix} dX \\ dY \end{pmatrix} \tag{5.9}$$

When m is located at the boundary between a dynamic obstacle and the floor, Equation 5.9 is not satisfied. Therefore, we can regard Equation 5.9 as a Classification Equation (CE), that is, the floor boundary point m can be classified as a stable obstacle or a dynamic one by confirming whether Equation 5.9 is satisfied or not. Actually, Equation 5.9 includes a small error ε depending on an image resolution, which is ignored. The following conditions should be satisfied in order to regard Equation 5.9 as the classification equation.

1. Floor boundary points have to be located at the boundary between obstacles and the floor correctly in the image.

2. Floor boundary points have to be tracked correctly.

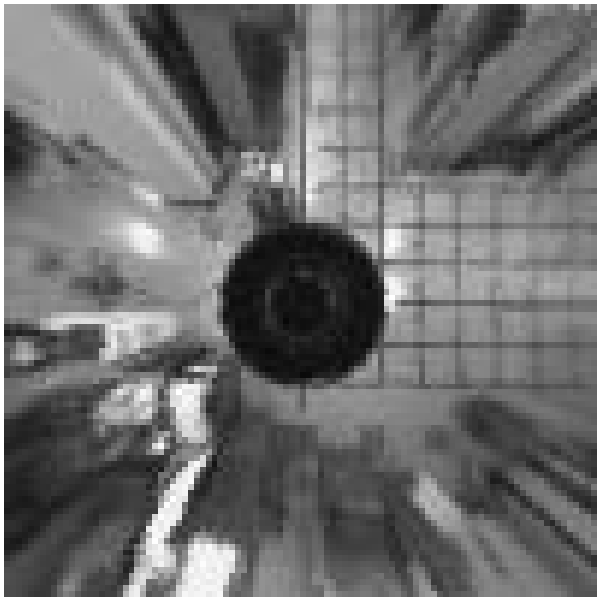


Figure 5.3: The bird's-eye image (5, Figure 3)

3. Camera parameters have to be decided correctly.

4. Odometry has to be calculated correctly.

Condition 4 is satisfied in the general environment, because the odometry is comparatively correct during short movement. Figure 5.2 verifies that parameters are not so bad that condition 3 is satisfied, too. Floor boundary points can be tracked easily and tracking is not a major problem when they are detected accurately, because they are located at the boundary where the colors change significantly. However, floor boundary points cannot always be detected correctly by using only the floor colors in various environments. We apply the result of confirming whether the CE is satisfied or not to the floor detection method.

5.2.2 Obstacle Classification

The CE is satisfied as long as floor boundary point m is located on the floor. One of the reasons why m is not located on the floor is that the threshold σ in Equation 5.5 is inappropriate. When the position of m does not satisfy the CE, σ is too large or m shows a dynamic obstacle. For confirmation, new floor boundary point m' is detected by decreasing σ in the direction where m is located to $\sigma - d\sigma$. The parameter $d\sigma$ should be small so that the robot does not narrow the floor area. The new floor boundary point m'

is tracked from t to $t - dt$ and classified by confirming the CE again. When the position of m' satisfies the CE, our robot regards m as a stable obstacle. Moreover, the position of m is changed to the position of m'. Conversely, if it is not satisfied, m is regarded as a dynamic obstacle. Our method changes the parameter dynamically by the result of the CE. For example, in Figure 5.4, the position of floor boundary point A located at the boundary between the floor and a static obstacle satisfies the CE. The point B that is not located at the boundary does not satisfy the CE. Therefore, B creates a new floor boundary point B' and B' is tracked from t to $t - dt$. Using the result of tracking, our robot confirms whether B' satisfies the CE or not. Because B' is located at the boundary, B' satisfies the CE in this case. Therefore, the position of B is changed to the position of B' and B is classified as a static obstacle. The point C located at the boundary between a dynamic obstacle and the floor also does not satisfy the CE. The point C creates a new point C' and its position is confirmed. Because $d\sigma$ is small, the point at the boundary does not create a new point far from the original point. The position of c' does not satisfy the CE in this case, and C is regarded as a dynamic obstacle.

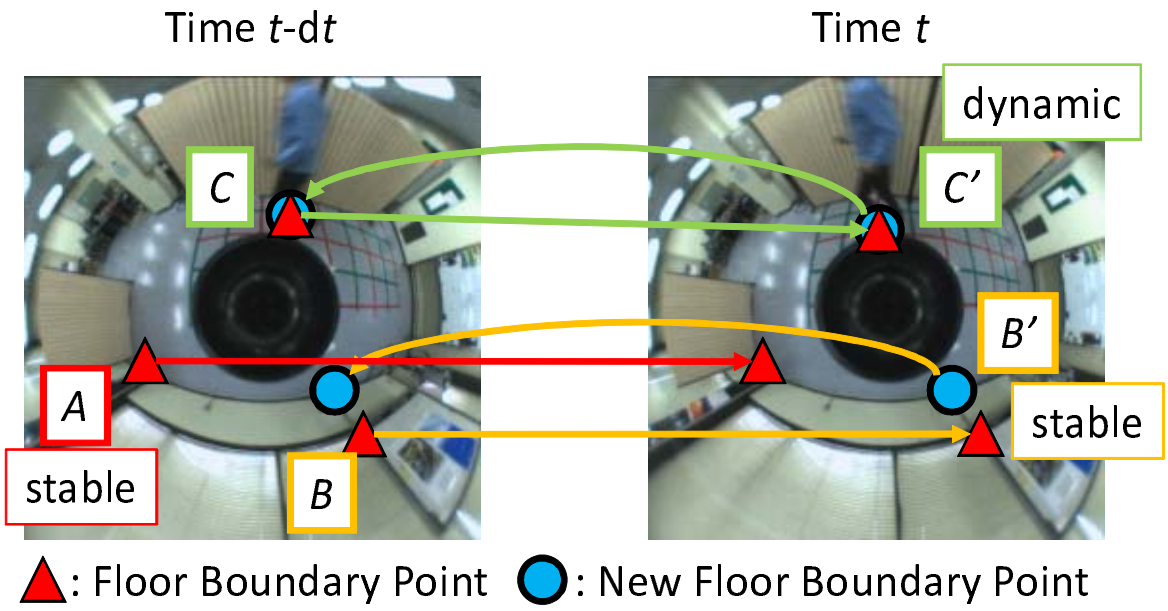


Figure 5.4: The example of the classification process by using floor boundary points (7, Figure 2)

If the threshold is low at the beginning of the robot's activation, all points are located on the floor. However, they are located between the boundary and the robot, and free

space looks very small. Our classification method first uses high thresholds and detects the boundary that is a little larger than the true boundary. Moving and confirming the CE refine the threshold of each direction where the floor boundary point classified as a dynamic obstacle is located. Finally, the robot adapts the threshold of each direction and makes it possible to locate and classify obstacles accurately.

5.3 Summary

In this chapter, we describe how to localize moving people by using a moving omnidirectional camera. We detect the floor by Ward's clustering and regard floor boundary points as obstacles. Our robots track them and compare movements of them with its odometry data. The points whose movements are different from the movement of the robot are classified as moving people. In order to classify accurately, tracking performs two times. When the movement of one point is different from the odometry, our robot strengthens dynamically a threshold to detect the point. Using the new threshold, the points are tracked again.

Chapter 6

Mobile Robot Self-Localization Based on Tracked Local Invariant Feature Points by Using an Omnidirectional Camera

This chapter defines tracked Local Invariant (LI) feature points and describes how the points are selected. This chapter also describes how our robot localizes its position by using tracked LI feature points.

6.1 Landmark Selsection from among Tracked Local Invariant Feature Points

6.1.1 Detection of Tracked Local Invariant Feature Points

The SURF method is used for detecting LI feature points. The SURF method performs faster than the SIFT method does. However, not all points detected by the SURF method can be tracked. For example, the SURF method can detect feature points at the center of wall and floor. The tracking method is not good at tracking such points. Moreover, some LI feature points change their feature vectors in the uncalibrated distorted omnidirectional image.

In order to avoid using LI feature points that are not tracked easily and change their feature vectors much, our robot moves along whole paths in the landmark selection phase. Our robot can move under ideal environments before shopping stores open. That is, we can assume our robot has enough time, knows the initial position, localizes its position using LRF and there are no people. While moving, our robot detects tracked LI feature

points while capturing continuous images. The top K LI feature points satisfying 6.1 and 6.2 for a long time are regarded as the tracked LI feature points.

$$|\mathbf{F}_G^{(t)} - \mathbf{F}_g^{(0)}| < T_{\mathbf{F}} \tag{6.1}$$

$$|\mathbf{x}_G^{(t)} - \mathbf{x}_{g'}^{(t)}| < T_{\mathbf{x}} \tag{6.2}$$

Here, g denotes one LI feature point. The robot started to move at time 0, G denotes one LI feature point detected at time t, and g' denotes a point obtained by tracking g until t. $F_i^{(t)}$ denotes a feature vector calculated by a SURF 64-dimensional feature vector of a point i at t. The members of the feature vector $F_i^{(t)}$ are normalized. A 2-dimensional vector $x_i^{(t)}$ denotes the position of a point i at t on image coordinates. An origin of image coordinates is defined as a center of the image. $T_{\mathbf{F}}$ and $T_{\mathbf{x}}$ are constant thresholds. Equation 6.1 enables the robot to detect an LI feature point that does not change the SURF feature vector very much while moving. Equation 6.2 enables the robot to detect the LI feature point close to a point obtained by tracking. When the tracking succeeds, the position of G is close to the position of g' in image coordinates. Therefore, it is easy to track g that satisfies both Equation 6.1 and Equation 6.2 for a long time and to detect it from various positions. The feature vectors of g do not change very much even in the distorted images.

6.1.2 Landmark Selection Based on Observation Positions

Tracked LI feature points detected by Equation 6.1 and Equation 6.2 are candidates of the landmark. Our robot regards tracked LI feature points whose positions on world coordinates are measured correctly as landmarks. Here, measuring correctly means that a distance between a measured position and correct position is less than a threshold T_d.

In order to measure a position (X_m, Y_m) of one tracked LI feature point m on world coordinates, m is measured from various positions. The robot can memorize an observation position $(X_{A1}, Y_{A1)}$ and a posture Θ_{A1} on world coordinates because the robot can localize its position under ideal environments in the landmark selection phase. From one position by using only one omnidirectional camera, the robot cannot measure (X_m, Y_m), but can measure a direction Θ_{I1} from the robot to m. When m cannot be tracked at t and similar tracked LI feature point m' is detected after t, Equation 6.1 is used in order to judge whether those two points are the same or not.

In order to measure (X_m, Y_m), a straight line L_{A1} passing through (X_m, Y_m) and (X_{A1}, Y_{A1}) is calculated by (X_{A1}, Y_{A1}), Θ_{A1} and Θ_{I1}. When m is observed from various observation positions $(X_{A2}, Y_{A2}), (X_{A3}, Y_{A3}), \ldots, (X_{An}, Y_{An})$, the lines $L_{A2}, L_{A3}, \ldots, L_{An}$ are also calculated. The point m is at the intersection of lines. However, not all lines intersect because observations often include errors. When the robot observes m from $(X_{A1}, Y_{A1}), (X_{A2}, Y_{A2}), \ldots, (X_{An}, Y_{An})$, the intersection is calculated by the least square error method as shown by Equation 6.3. Here, $\mathbf{M}^+$ denotes a para-inverse matrix of matrix $\mathbf{M}$.

$$\begin{pmatrix} X_m \\ Y_m \end{pmatrix}^{(t)} = \begin{pmatrix} \tan(\theta_{A1} + \theta_{I1}) & -1 \\ \tan(\theta_{A2} + \theta_{I2}) & -1 \\ \vdots & \vdots \\ \tan(\theta_{An} + \theta_{In}) & -1 \end{pmatrix}^{+} \begin{pmatrix} X_{A1}\tan(\theta_{A1} + \theta_{I1}) - Y_{A1} \\ X_{A2}\tan(\theta_{A2} + \theta_{I2}) - Y_{A2} \\ \vdots \\ X_{An}\tan(\theta_{An} + \theta_{In}) - Y_{An} \end{pmatrix} \tag{6.3}$$

The point M is at the intersection of the lines only when the point is observed accurately from various observation positions, as shown in Figure 6.1. In Figure 6.1, the line arrow denotes the direction (posture) of the robot.

An example of a measured landmark is shown in Figure 6.2. Figure 6.2 shows 4 omnidirectional images taken from 4 observation positions. The landmark is the center point of a dotted line circle in each image. The same place is detected as the landmark from various observation positions.

(X_m, Y_m) is calculated accurately by the least square method when the same landmark is detected accurately from various observation positions as seen in Figure 6.2. However, the calculation sometimes fails. For example, when there are many similar things and the robot regards different objects as the same, the robot cannot calculate the position of the intersection, as shown in Figure 6.3. Additionally, when observations do not fail but the observation positions are close to each other, the intersection cannot be calculated.

Considering a relationship between observation positions and the accuracy of the landmark measurement, tracked LI feature points that may fail to be measured are deleted. Tracked LI feature points that can be measured correctly are regarded as landmarks. Here, we use Support Vector Machine (SVM) in order to classify the points as landmarks or not.The SVM is good at classifying 2 classes. It is often used for classifying 2 classes and resolves various problems of classifying [89][90][91]. The SVM classifies correctly by deciding boundaries between one class and the other. It decides boundaries by maximizing distances between data and the boundaries. It can decide non-lineear boundaries by

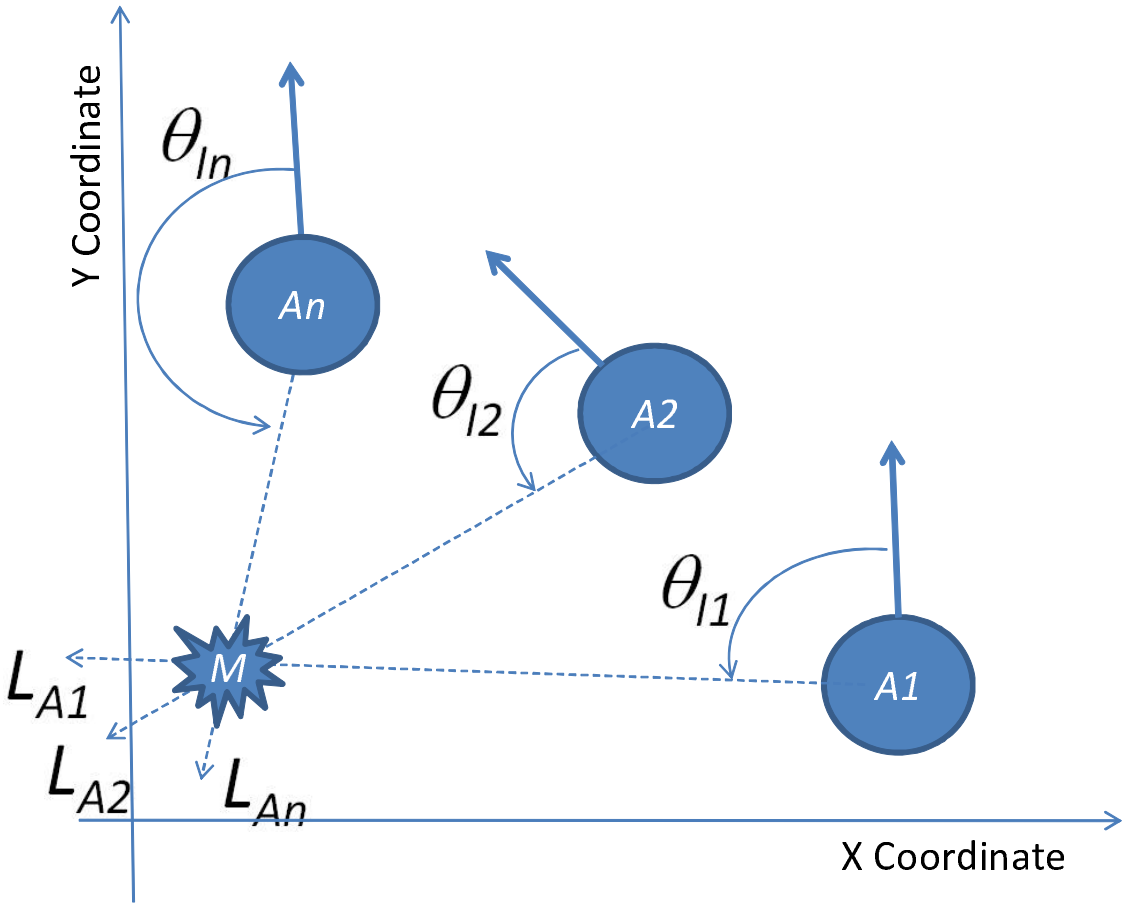


Figure 6.1: Example of successful measurement of the position of the LI feature point m (7, Figure 3)

the kernel function $K(\mathbf{x}_1, \mathbf{x}_2)$. The kernel funcion projects feature vectors $(\mathbf{x}_1, \mathbf{x}_2)$ to a high-dimentional space and the non-linear boundaries change the linear boundaries in the space.

6-dimensional vector is used as a feature vector for SVM. The feature vector consists of 6 feature quantities as follows:

1. the number of observations
2. the least square error of the measurement [rad^2]
3. the average distance from the robot to the points in the image [pixel]
4. the distribution of the observation positions' x-coordinate [m^2]

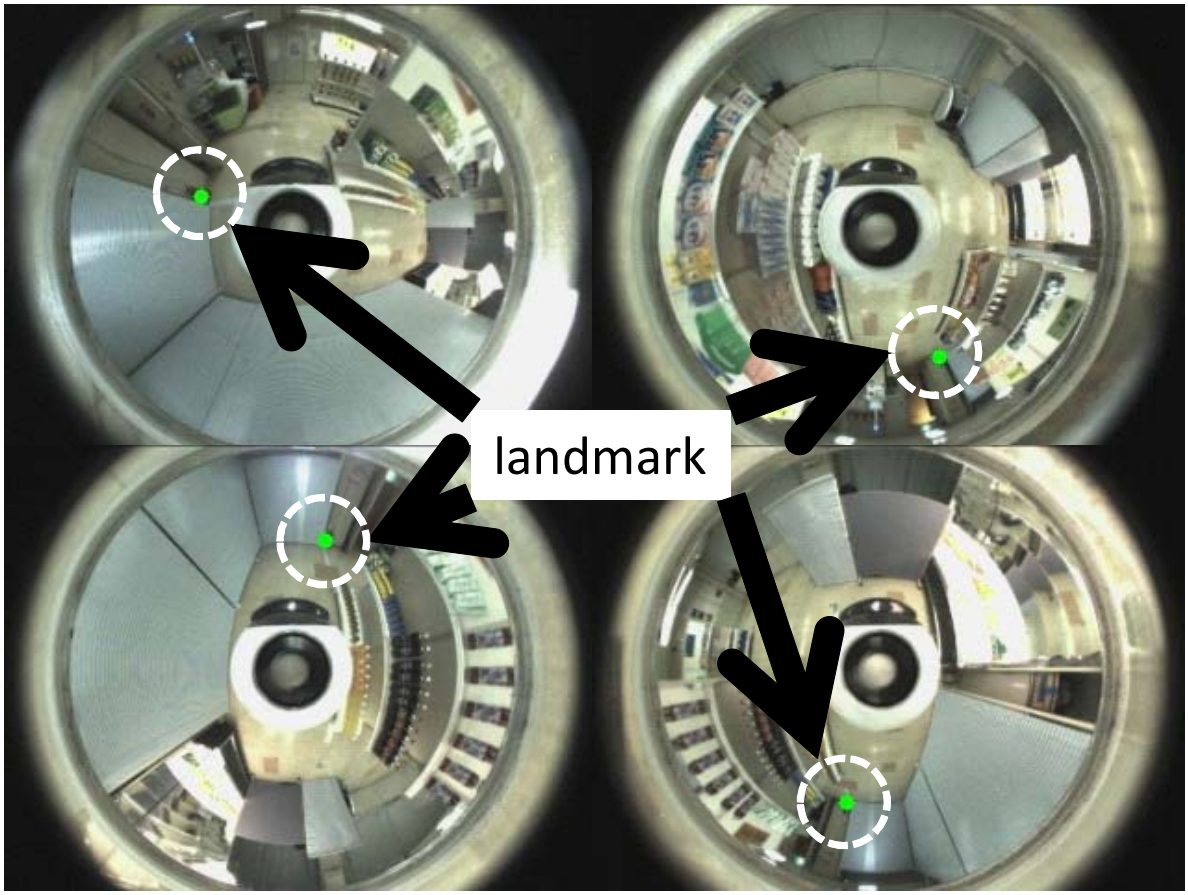


Figure 6.2: The landmark from various observation positions captured by an omnidirectional camera (7, Figure 4)

5. the distribution of the observation positions' y-coordinate [m^2]

6. the distribution of the directions from the robot to the points [rad^2]

These 6 feature quantities relate to the observation positions and postures. At the end of the landmark selection phase, the feature vectors of all tracked LI feature points are calculated. The vectors are calculated by the result of the measurements and the information of the observation positions. SVM that has already been trained by training data classifies the points as landmarks or not. In order to make the training data, we first make the robot move to another place and obtain N landmark candidates. Next, we manually label the candidates as landmarks or not by comparing measured positions with correct positions of landmarks. This manual step can be omitted once SVM has been trained.

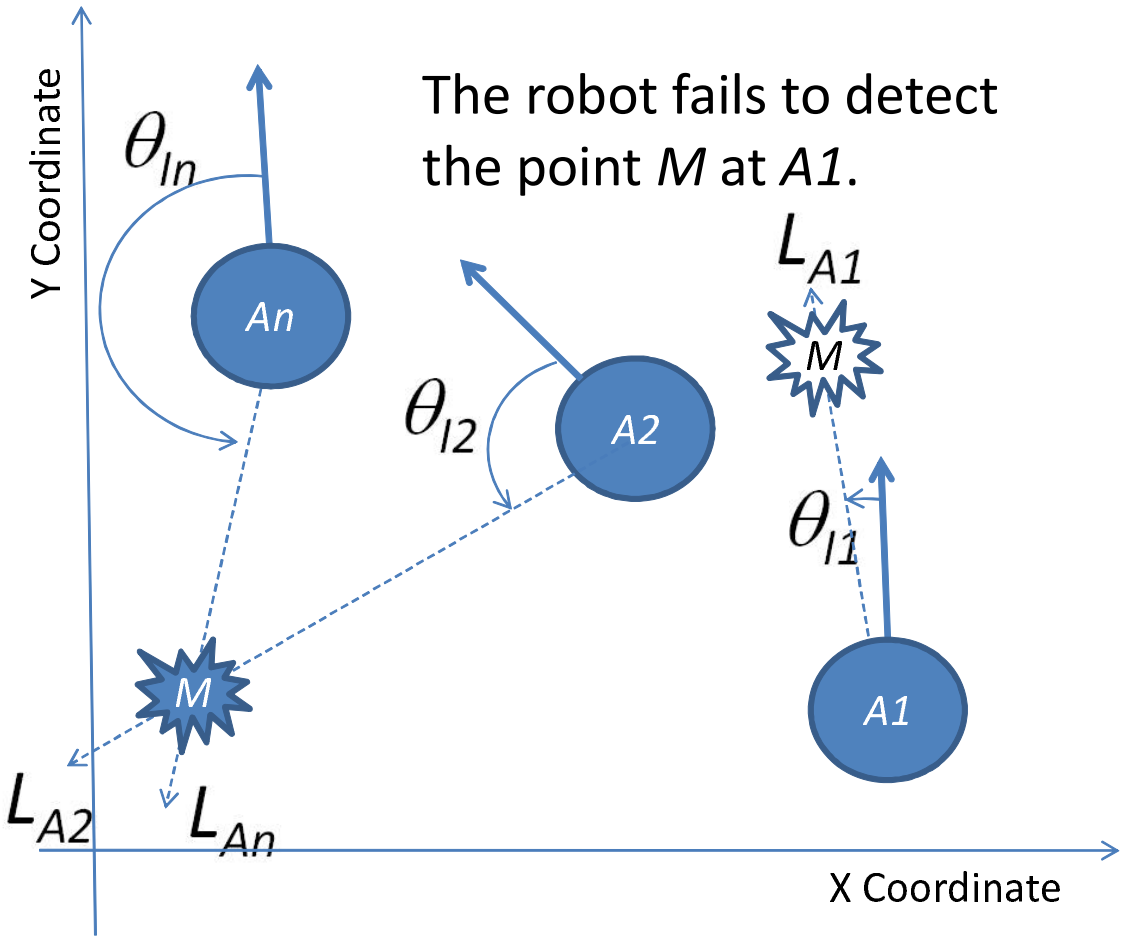


Figure 6.3: The example of failure in measuring the position of the LI feature point M (7, Figure 5)

6.2 Self-localization Based on Tracked Local Invariant Feature Points

6.2.1 Tracking Error Criterion

While the robot moves, it localizes its position while taking an omnidirectional image continuously. The landmark is detected from the continuous images. The position and posture of the robot can be calculated by detecting more than 3 landmarks. In order to realize the fast self-localization,once the robot detects landmarks by using the SURF method that performs slowly, the landmarks are tracked fast continuously. It is easy to track the landmarks for a long time because the landmarks are tracked LI feature points. However, tracking landmarks will certainly fail. For example, when people move between

the robot and the landmarks, targets of the tracking method change. As long as the robot does not know that the targets change, self-localization fails and the robot continues to track different targets.

We propose a tracking error criterion calculated by the directions to the tracked landmarks in the uncalibrated omnidirectional image. Our robot stops to track the landmarks that have a high tracking error criterion. Stopping tracking minimizes the self-localization error.

The tracking error criterion E is defined as Equation 6.4, Equation 6.5 and Equation 6.6. Here, (x_i, y_i) and (x'_i, y'_i) denote the position on world coordinates and image coordinates of landmark i, respectively. (X, Y, Θ) denotes the robot's position and posture on world coordinates estimated by using more than 3 landmarks.

$$E = |\alpha_{i'} - \beta_i| \tag{6.4}$$

$$\alpha_{i'} = \tan^{-1} \frac{y_{i'}}{x_{i'}} \tag{6.5}$$

$$\beta_{i'} = \tan^{-1} \frac{y_i - Y}{x_i - X} - \Theta \tag{6.6}$$

$\alpha_{i'}$ denotes the direction to the landmark i that can be measured by an omnidirectional image. β_i denotes the direction to the landmark i that can be calculated on world coordinates. Figure 6.4 shows a relationship between $\alpha_{i'}$ and β_i. As shown in Figure 6.4, when both self-localization and detecting landmarks succeed, $\alpha_{i'}$ is equal to β_i. When $\alpha_{i'}$ is equal to β_i, the tracking error criterion E is equal to 0 which is the smallest value.

On the other hand, when either self-localization or detecting landmarks fails, E increases. For example, when the tracked target changes from the landmark to the person who walks between the robot and the tracked landmark, $\alpha_{i'}$ differs from β_i gradually.

6.2.2 Self-localization System Switching between SURF Method and Tracking Method

Our self-localization system is shown in Figure 6.5. "Sub Flow" in Figure 6.5 denotes a process path that is passed in the case that the tracking error criterion E is high. In "Sub Flow", landmarks are detected slowly by the SURF method. "Main Flow" denotes a process path that is usually passed. In "Main Flow", landmarks are detected fast by the tracking method. The number of passing "Main Flow" is more than that of "Sub Flow". Once our system detects landmarks through "Sub Flow" and our robot localizes

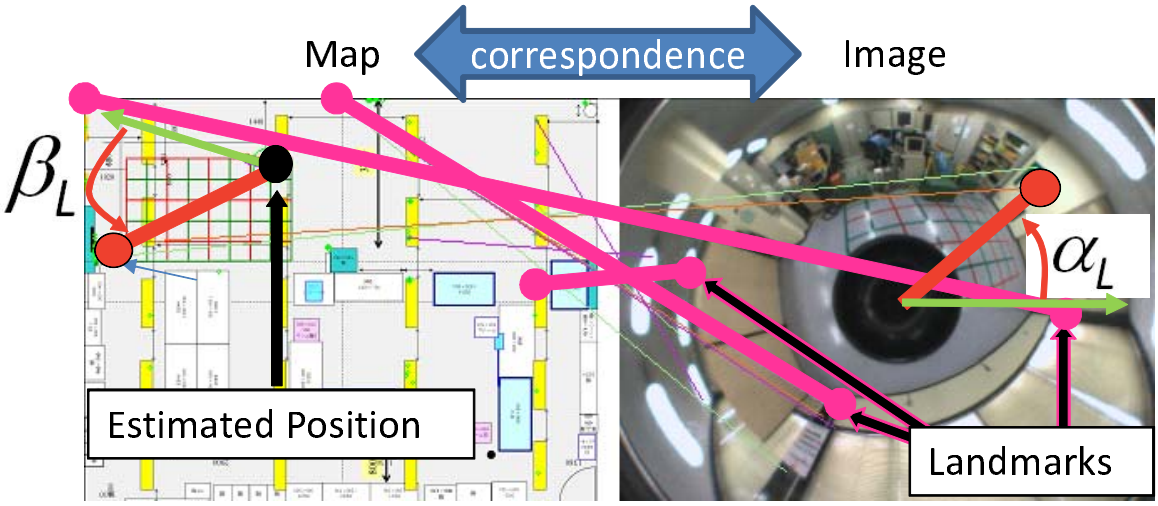


Figure 6.4: The relationship between $\alpha_{i'}$ and β_i (7, Figure 6)

its position, our robot localizes its position by using the tracking method through "Main Flow". Our robot also removes the landmark that has the high E while moving. When the number of landmarks is less than a threshold, our robot detects landmarks again by using the SURF method. Details of the self-localization steps are as follows:

1. The robot captures an omnidirectional image.
2. Self-localization is performed through "Sub Flow".
3. The tracking error criteria E of all landmarks are calculated.
4. The landmark whose E is higher than a threshold Tp is removed.
5. If more than Tn landmarks exist in step 4, self-localization is performed through "Main Flow" after the next omnidirectional image is captured. Otherwise, self-localization is performed through "Sub Flow".
6. The robot repeats the process from step 3 to step 5.

6.3 Summary

In this chapter, we describe how to detect landmarks fast and localize the robot accurately by using them in the omnidirectional image. In order to realize fast and accurate self-localization, we develop tracked scale and rotation invariant feature points. We regard

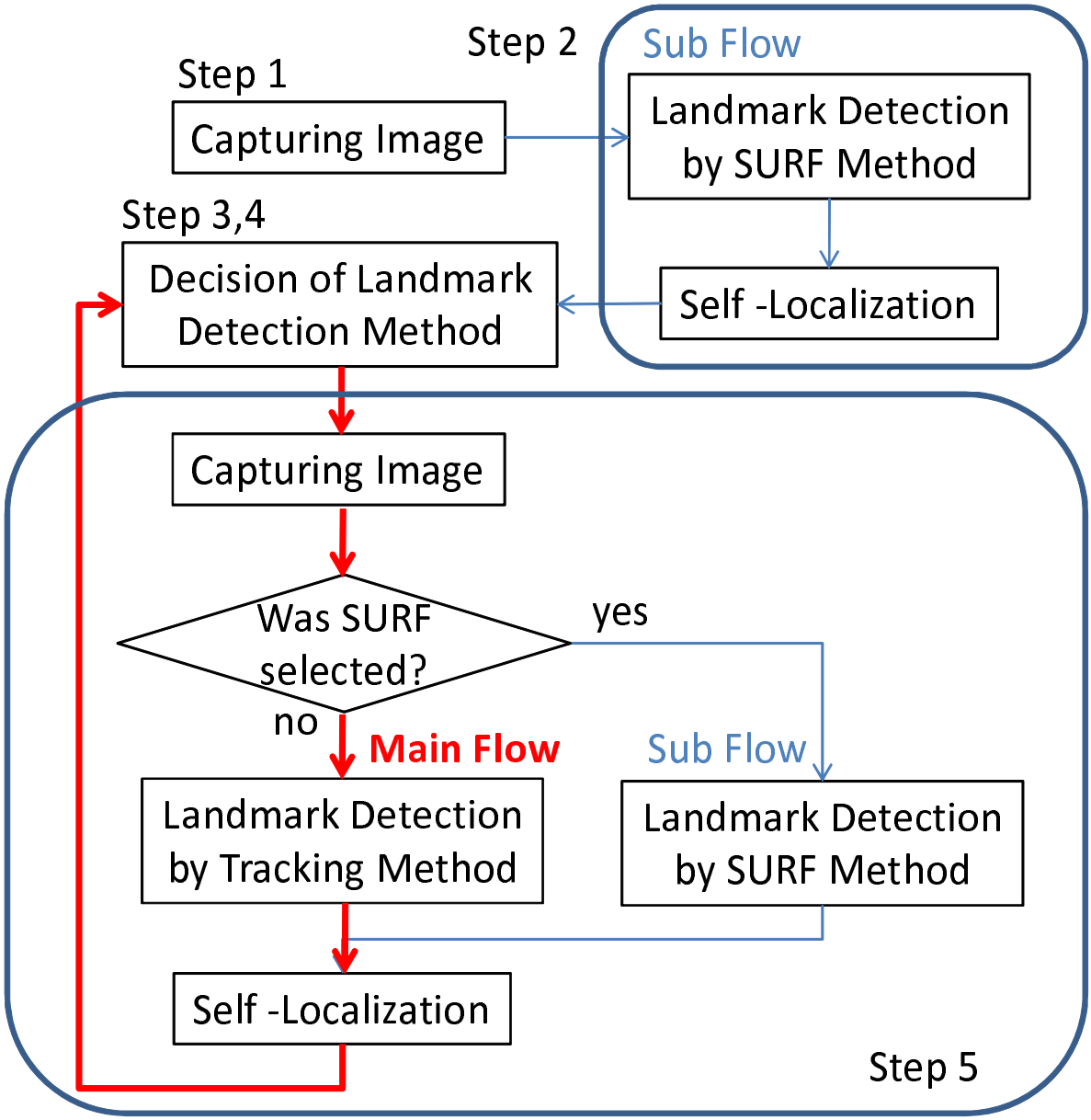


Figure 6.5: The developed self-localization system (7, Figure 7)

the points as landmarks. The points are detected by using both the slow SURF method and the fast tracking method. Before the robot provides services, it obtains continuous omnidirectional images and obtains feature points by SURF. We select the points which can be tracked for a long time from among the feature points. The selected points denote the tracked scale and rotation invariant feature points. Once the robot detects the landmarks by SURF, the robot tracks them fast and localize itself fast and accurately.

Chapter 7

Implementation and Evaluation of Detecting People and Objects for Navigation-Transportation Robot

This chapter introduces our navigation and transportation robot ApriTau^{TM}. This chapter also describes an ability of our floor detection method, accuracy of our classification method and self-localization method by ApriTau^{TM} at a mock shopping store.

7.1 Design of the Navigation System

7.1.1 Navigation Robot ApriTau^{TM}

Hardware of ApriTau^{TM}

Our obstacle classification method and the self-localization method are implemented on our robot called ApriTau^{TM}, as shown in Figure 7.1. It is 650 [mm] wide and 1200 [mm] tall. Its weight is approximately 75 [kg]. It has Li-ion batttery and works for 1 [h] continueously. It has a vehicle that can acquire odometry data and moves at 0.6 [m/s]. The vehicle consists of independent 2-motor drive. The motor is made by MAXSON. The accuracy of the odometry data is 1.0 [%], that is, the error in the odometry data is 1.0 [cm] when AppriTau^{TM} moves by 1.0 [m]. An omnidirectional camera is mounted on top of its head and does not move with the head. The omnidirectional camera consists of a mirror and a camera. The mirror is VS-C450u-200-TK made by VSTONE. The elevation angle and the depression angle of the omnidirectional camera are 15 [deg] and 55 [deg], respectively. By taking images while moving, it synchronizes the odometry data. ApriTau^{TM} takes images with a size of 320×240 [pixels] continuously at 30 [fps] by the camera. The camera is the flea2 made by Point Grey Research.

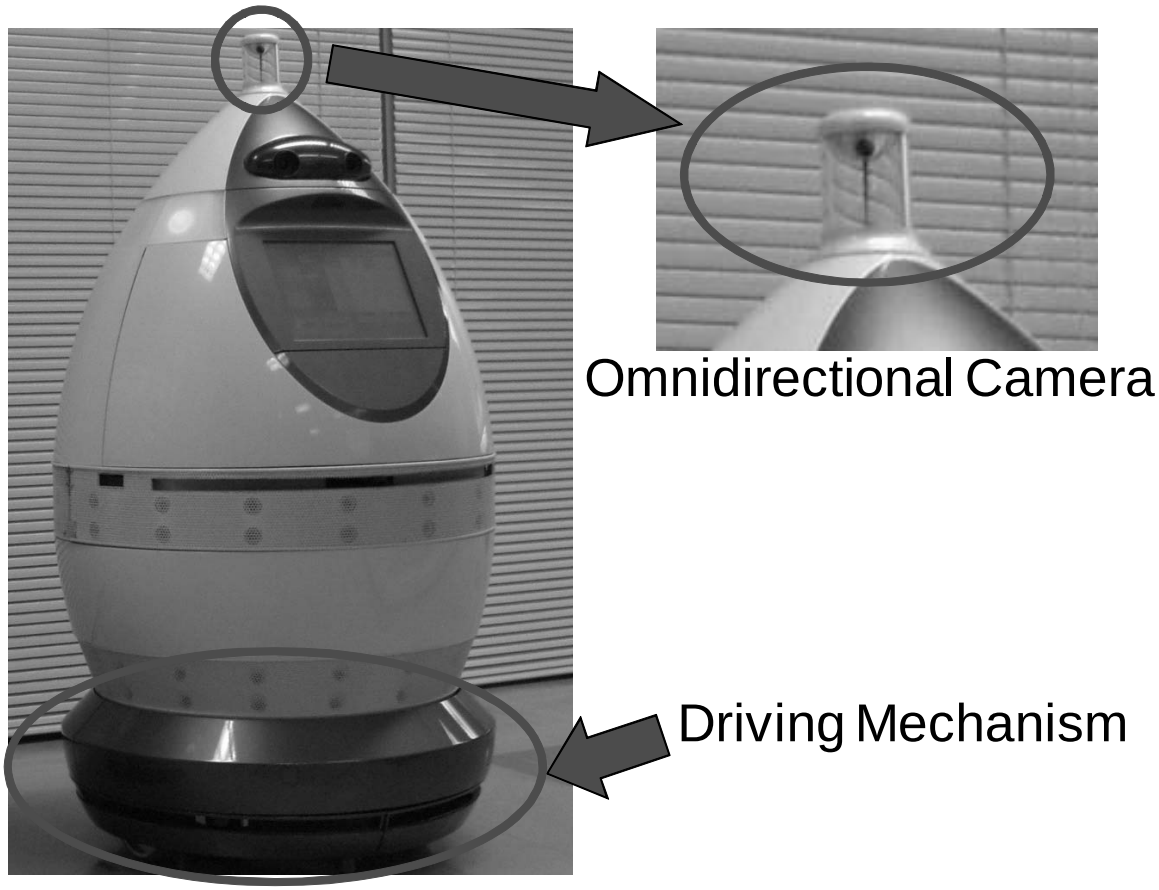


Figure 7.1: The navigation robot ApriTau^{TM} (5, Figure 6)

ApriTau^{TM} cooperates with the system installed in the shopping store. The shopping store is equipped with some mobile robots that move around in the store, multiple fixed camera systems that can locate persons and robots, and a server system (RT-server) to manage position information of the persons and the robots. There are two kinds of robot (navigation and cart). In this thesis, we focus on the navigation robot ApriTam^{TM}. The cart robot has been reported in another paper [92][93][94].

The system construction of the robot and the store is shown in Figure 7.2. Four PCs, PC1 - PC4, are mounted in the robot. PC1 executes motion control. PC2 and PC3 are dedicated to capture and process images of the cameras. PC4 is responsible for human interface processing and higher thinking processes such as total planning of robot behavior. The avoidance motion is mainly related to the performance of PC1. PC1 is equipped with an Intel Pentium M (2.0 GHz) processor and 2.0 [GB] memory. PC1 is managed by a real-time OS (ART-Linux). The obstacle classification and self-localization are related to

the performance of PC2 and PC3. PC2 and PC3 are equipped with an Intel Core2 Duo (1.4 [GHz]) processor and 2.0 [GB] memory. PC2 and PC3 are managed by a Windows XP. The robot refers to the information accumulated in the server when the robot plans its action. The communication system between the robot and the environment system is constructed using a wireless LAN.

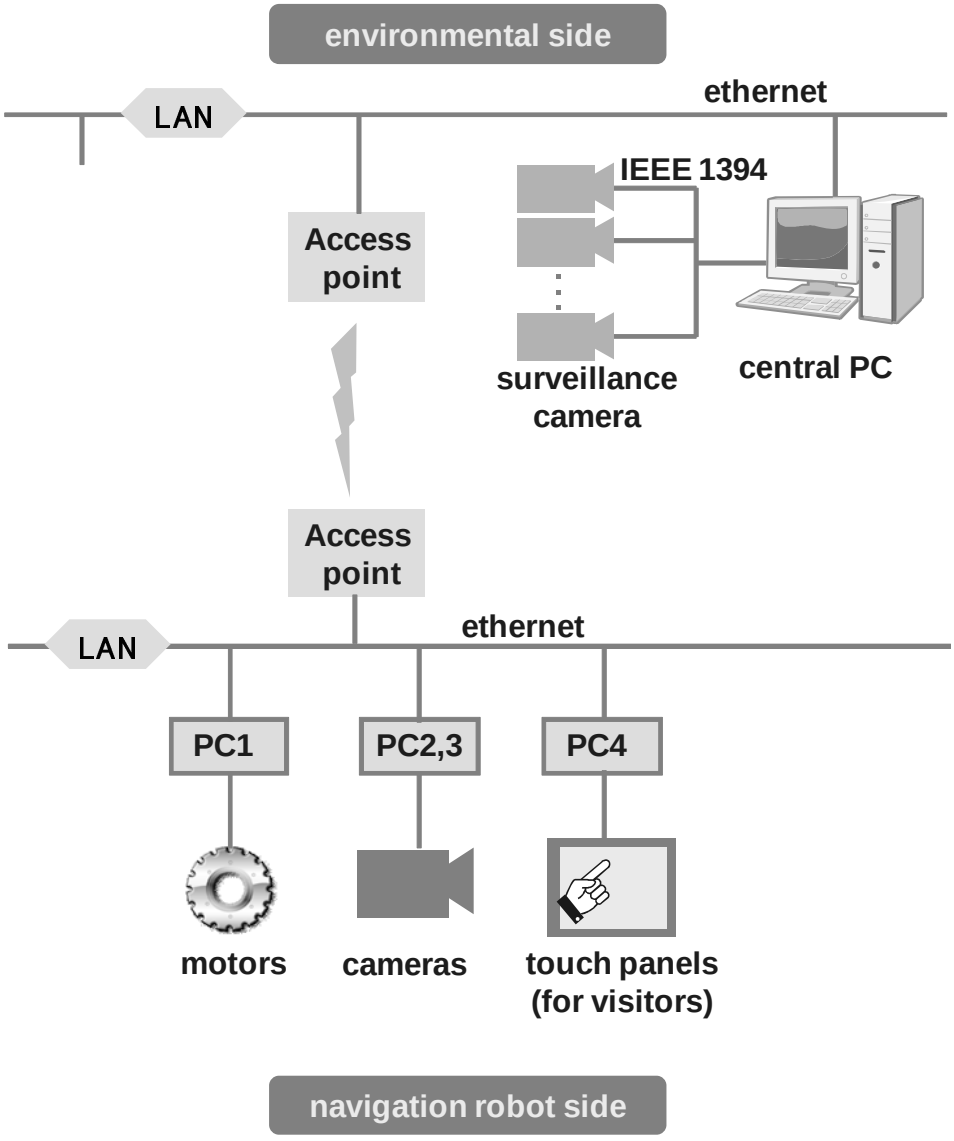


Figure 7.2: The system construction of the mock shopping store

Software of ApriTau^{TM}

ApriTau^{TM} has also special functions except the obstacle classification and the self-localization for moving. One of the most essential functions is an collision avoidance function [95]. From the viewpoint of robustness, a reflective collision avoidance algorithm has been used with predefined map-based path planning. However, because the algorithm is composed of simple operation rules, a robot depending on the reflective collision avoidance algorithm may often move inefficiently. To ensure efficient collision avoidance, we develop a reflective collision avoidance technique that corrects direction and magnitude of robot velocity independently using 1-dimensional potential-like functions ruled by a pseudo distance that modifies its own distance in proportion to an angle from the robot traveling direction to the obstacle direction. Moreover, two potential functions refer to the anisotropic pseudo distance that is transformed by the magnitude of the relative directional angle from a traveling direction to an obstacle. With these characteristics, an effective collision avoidance algorithm is achieved that generates smooth corrective motion and works only if an obstacle appears in a collision course. Next, by experiments involving movement using an actual robot, the developed technique is confirmed to work efficiently in handling various dynamic obstacles. Therefore, because the robot can search objects around it quickly and independently of the path planning system, the developed technique is effective from the viewpoint of the improvement of the safety of the mobile robot. Further, because the collision avoidance algorithm can be executed quickly, namely, in a 1.0 [ms] periodic motion control system loop, the technique is effective from the viewpoint of calculation cost.

ApriTau^{TM} uses the pyramidal implenentation of the Lucas Kanade feature tracker [96] (LKT) as the tracking method. Previous works said the method works well even in the continuous omnidirectional camera images captured by a mobile robot [97]. LKT method can realize to track points fast because it solves system of equations that assume there are just small changes around the tracked points. The pyramidal implementation improves the LKT method to track the points which change very much. In order to track them, the pyramidal implementation uses multiple pyramidal images wihch have different resolutions from high to low and different captured area from narrow to broad around the tracked points. The pyramidal implementation applies the LKT method to pyramidal images in order of resolution from broad to narrow. Using the result of traking points

in the broad image, the pyramidal implementation can track the points that change very much.

In this system, the entire motion control system of the robot consists of components on the Robot Technology Middleware (RTM) [98] because it is easy to apply them to other robots. RTM is a common platform that integrates modules by using networks. The modules of RTM are called Robot Technology Component (RTC). Here, we prepared capturing images RTC, obstacle location classification RTC, self-localization RTC, motor control RTC, and executing scenario RTC. We describe how the robot moves in the executing scenario RTC. RTC is connected as shown in Figure 7.3. RTC has 3 kinds of ports, an input data port, an output data port and a service port. The triangle on the left side of the component , the triangle on the right side and the square fixed on RTC in Figure 7.3 denote the input data port, the output data port and the service port, respectively. The output data port provides continuous data. The input data port receives continuous data. The service port provides data or services when RTC requests the port. In this system, the capturing images RTC provides continuous 320×240 [pixels] image data. The motor control RTC provides odometry data when either the self-localization RTC or the obstacle classification RTC requests. The obstacle classification RTC provides the positions of obstacles continueously. The self-localization RTC provides the positions of ApriTau^{TM} continuously. The executing scenario RTC receives the position data and asks the motor control RTC to drive the motors.

7.1.2 Design of Obstacle Classification

Figure 7.4 shows our classification system in detail. ApriTau^{TM} takes images continuously and inputs them to the system. In image at $t - dt$, the system detects 360 floor boundary points using the result of tracking previous points or the floor detection method. In the case of tracking points on the boundary, the tracked point moves a little along the boundary. Therefore, we confirm tracking result every 1 degree in the omnidirectional image. If there are more than two points in the same direction, we select closer one. Moreover, if there are no points in the direction, we detect the new point in the direction as a stable obstacle.

Figure 7.5 shows the example of the output. Dark gray or light gray points are floor boundary. 360 points are detected every one degree. If we use more points, they form complete floor boundary. However, we think 360 points are sufficient for the robot's

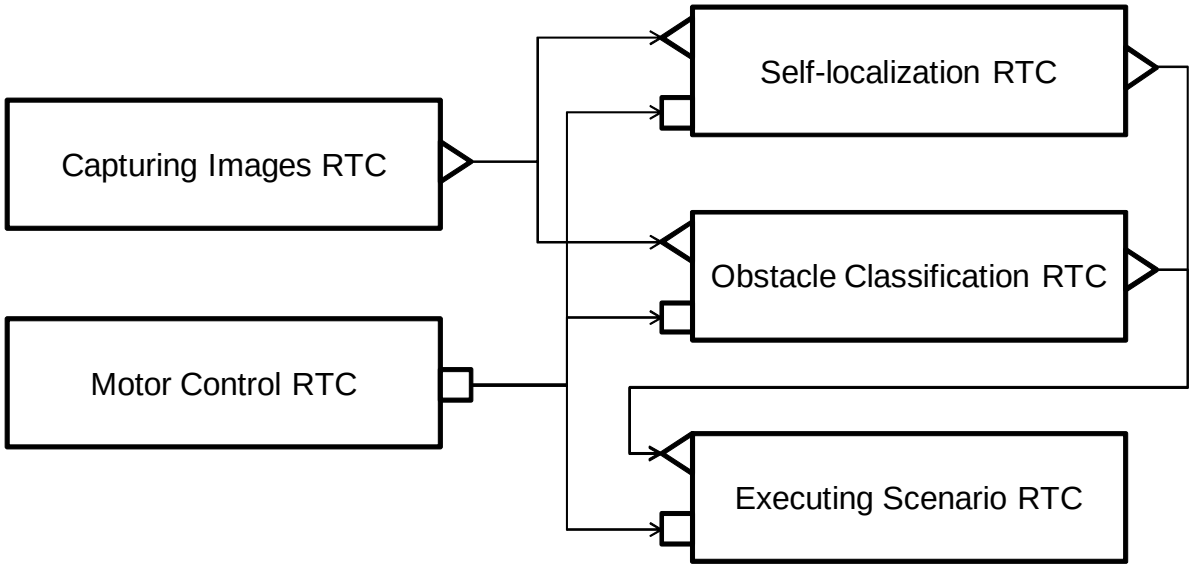


Figure 7.3: The connection of RTC

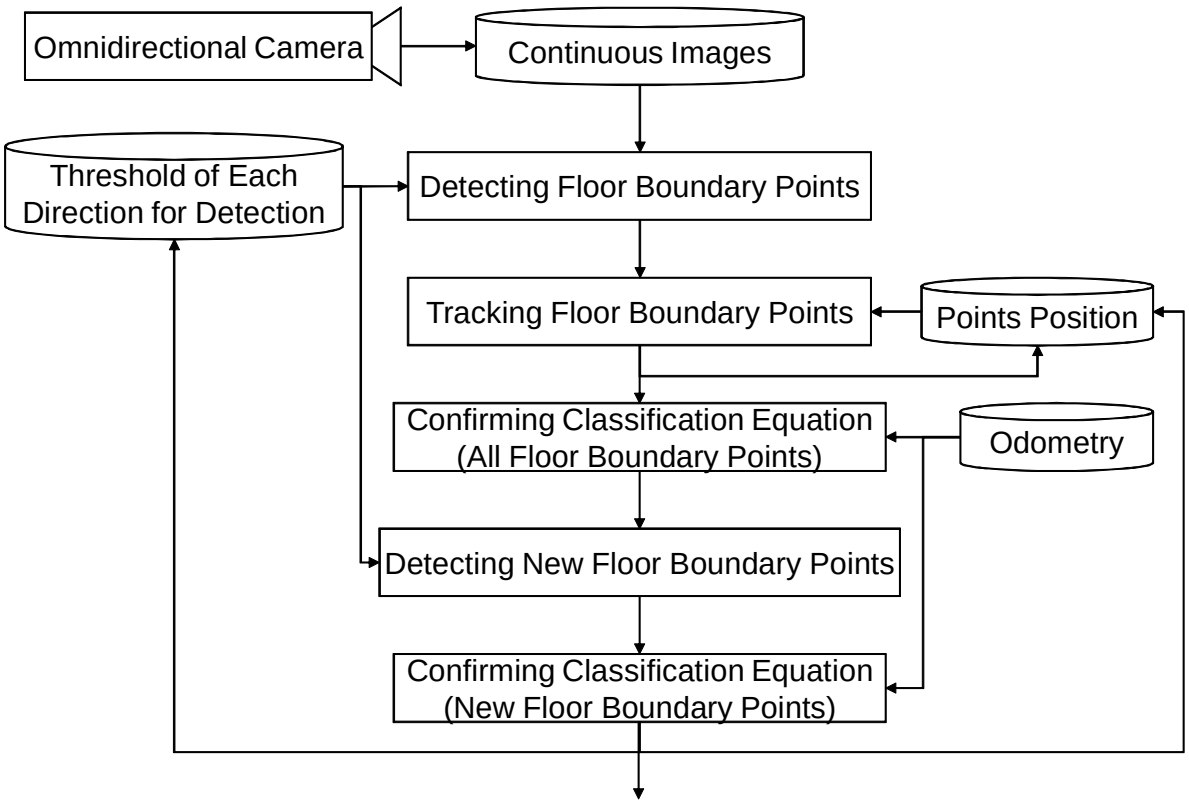


Figure 7.4: The classification system in ApriTau^{TM} (5, Figure 7)

movement. These points are tracked and classified. In Figure 7.5, light gray and dark gray points are classified as stable obstacles and dynamic obstacles, respectively. Most of them are located at the boundary between the floor and obstacles. A line is drawn from the image center to the average of dark gray points' positions. This system integrates floor boundary points which are classified as dynamic obstacle like the line, when points which are classified as dynamic obstacles are located near the other points which are classified as dynamic obstacles.

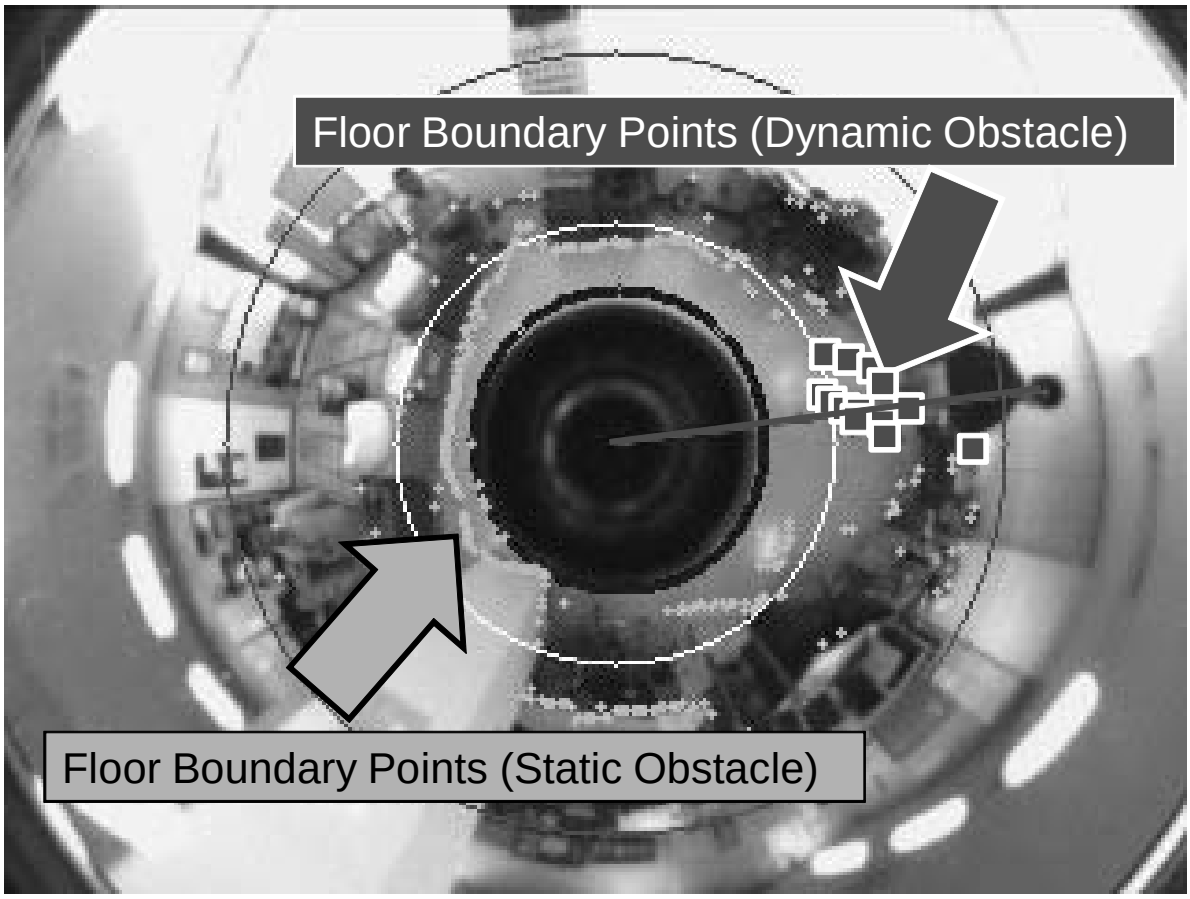


Figure 7.5: The output of classification system (5, Figure 8)

Here, the threshold T_{CE} (mentioned in Chapter 5) is 20 [cm]. dt is about 200 [ms]. σ is initialized by 10 and $d\sigma$ is 3. The initial values of σ and $d\sigma$ are decided experimentally.

7.1.3 Design of Self-localization

The system related to the self-localization is shown in Figure 7.6. The function uses same images used by the obstacle classification function. ApriTau^{TM} makes a landmark database before opening the shopping store. The database has both a location data on the shopping store coordinate and the SURF 64-dimensional feature vector.

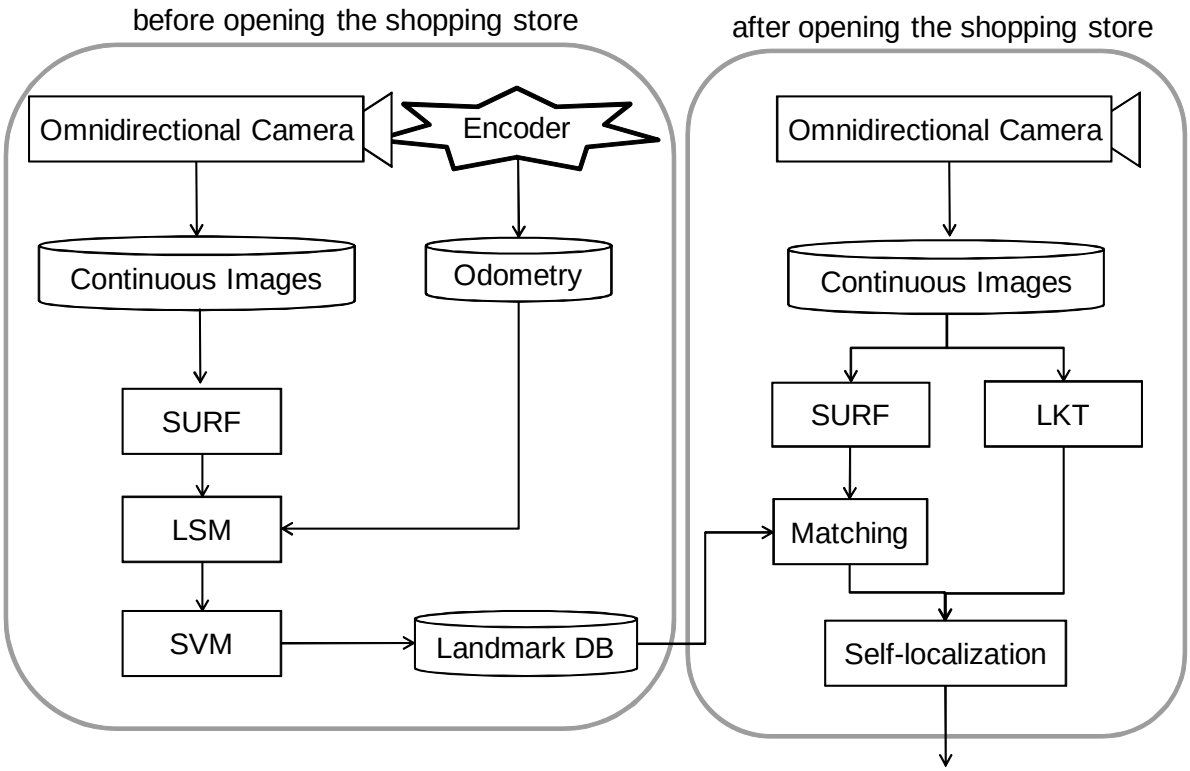


Figure 7.6: The self-localization system

The SVM is trained by 100 tracked LI feature points. For previous settings, we make training data in an experimental room. Training data consist of two kinds of feature vectors. One is the feature vector which discripts a landmark located correctly by using the method showed by in Chapter 6. The other is the feature vector which discripts a landmark located incorrectly by using the method. Here, the threshold T_d is 60[cm]. We decide the threshold based on a size of a small shelf whose length is approximately 60 [cm]. We use the Gaussian kernel which is the most general kernel function of all function as shown in Equation 7.1

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\frac{\|\mathbf{x}_1, \mathbf{x}_2\|}{2\sigma^2}) \tag{7.1}$$

The parameter σ is decided after an evaluation in Chapter 7.

In this system, the thresholds $T_{\mathbf{F}}$, $T_{\mathbf{x}}$ and the parameter K as shown in Chater 6 are 40, 1.5 and 9, respectively. The thresholds T_p and T_n are 5 [deg] and 5 points respectively. These thresholds and parameters are decided experimentally, considering the resolution of the image and the size of the mock shopping store. For the self-localization, we use the pyramidal implementation of LKT as same as the classification method does.

7.2 Evaluation of Floor Detection by ApriTau^{TM}

7.2.1 Aim and Sequence of Floor Detection Evaluation

In order to evaluate the floor detection capability of our method, we compared our method with the previous floor detection method based on the GMM.

We performed experiments at a mock shopping store, as shown in Figure 7.7. The size of the mock shopping store was 10 [m] by 10 [m]. There were some shelves, a refrigerator, and a self-checkout machine at the mock shopping store. Each shelf had a sign that showed the category of products and real products classified in the category. The floor had two simple colors.

Figure 7.7: The experimental room used as a shopping store

Two datasets were used in this experiment. Each dataset consisted of learning data and test data.

First, color distribution is learned and parameters such as the number of mixed Gaussian and the threshold are optimized for the GMM by using the learning data of the first dataset. Our method learns color distribution automatically, and parameters such as T_N, T_D, and σ discussed in Chapter 5 are optimized manually. Using the optimized parameters and learned colors, both methods are tested by using the test data of the first dataset.

Next, both methods learn only color distribution by using the learning data of the second dataset. Using the learned colors and parameters optimized by using the first dataset, both methods are tested by using the test data of the second dataset. The second dataset is obtained in various places.

In both test cases, the values of two parameters (Hit and Correct Rejection (CR)) are calculated for evaluation. P_f denotes the number of pixels to which the floor is projected. P_{cf} denotes the number of pixels detected correctly as the floor. P_o denotes the number of pixels to which objects except the floor are projected. P_{co} denotes the number of pixels detected correctly as objects, except the floor. Although pixels that are close to the robot are more important than pixels far from it for the robot's smooth movement, in this experiment we regard all pixels as equal because we want to evaluate only the floor detection capability.

$$Hit = \frac{P_{cf}}{P_f} \tag{7.2}$$

$$CR = \frac{P_{co}}{P_o} \tag{7.3}$$

7.2.2 Results

The outputs of the first test and the second test are shown in Figure 7.8 and 7.9. In the figures, (a) and (b) are outputs of the GMM method and our method, respectively. The figure also shows that the GMM cannot detect the floor which is far from our robot. White and black regions correspond to the floor and the objects except the floor, respectively. The calculated evaluation values are shown in Table 7.1.

These results show that, when parameters are optimized manually, the ability of the GMM floor detection method is similar to that of our method. However, when parameters

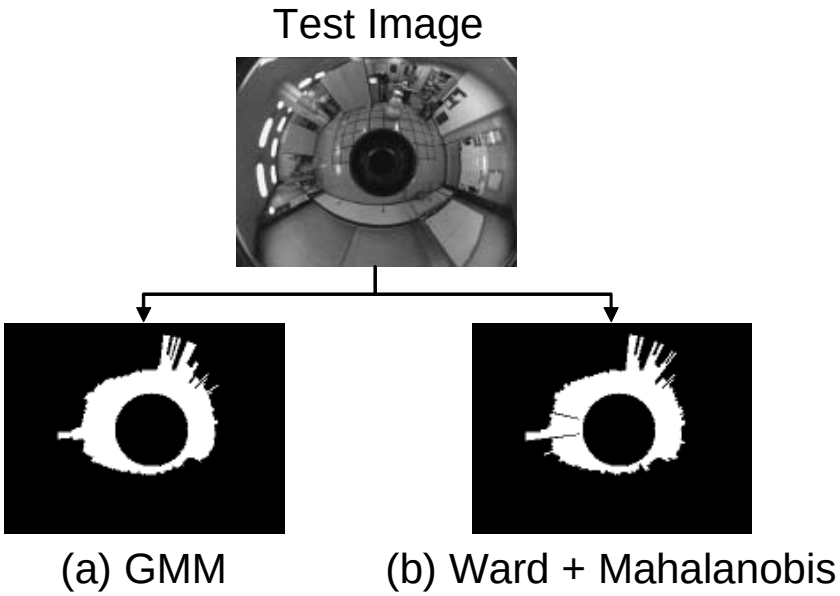


Figure 7.8: The output of the first test (5, Figure 9)

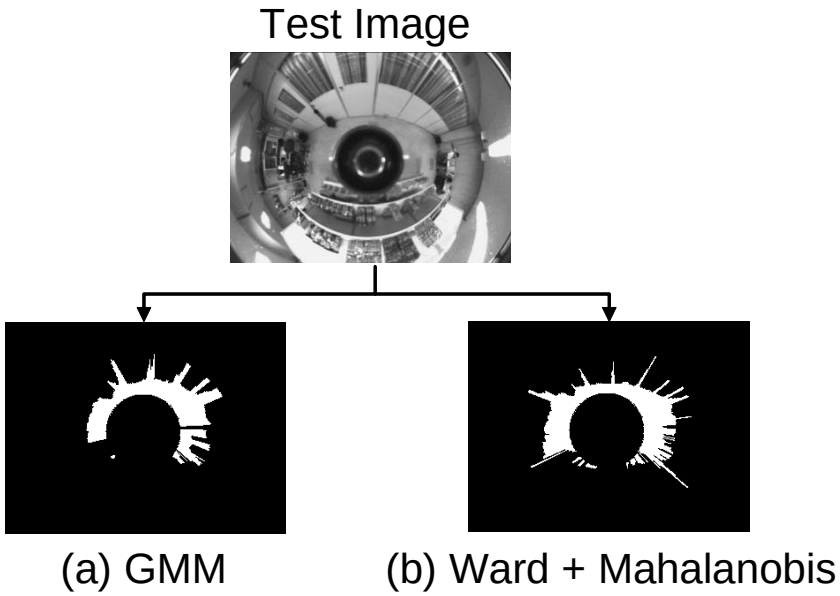


Figure 7.9: The output of the second test (5, Figure 10)

Table 7.1: Comparing the GMM method with our method.

Method	Hit	CF
First Test the GMM (optimized parameters)	0.95 (7277 / 7655)	0.95 (65826 / 69145)
First Test Ours (optimized parameters)	0.94 (7210 / 7655)	0.96 (66029 / 69145)
Second Test the GMM (first test parameters)	0.73 (13204 / 18022)	0.96 (203953 / 212378)
Second Test Ours (first test parameters)	0.90 (16245 / 18022)	0.95 (202061 / 212378)

are not optimized, the Hit value of the GMM method is lower than that of our method. The result shows both maximum abilities are similar, but our method learns the floor colors more easily than the GMM does.

7.3 Comparing Developed Obstacle Classification with Relational Methods

7.3.1 Aim and Sequence of Experiment

In order to confirm the effectiveness of changing the threshold σ dynamically based on the result of the CE, we compared the classification ratio of our method with that of a simple method using a constant threshold and the previous method. In this experiment, we use the previous metohd that changes the omnidirectional image to the non-distorted image and detects positions where the optical flow is different from an average of all optical flows in the image. The experimental steps are as follows:

1. ApriTau^{TM} and another robot move on the given route.

2. ApriTau^{TM} takes images synchronized with odometry data continuously while moving.

3. The images and the odometry data are input to the systems of both our method and the simple method. Note that, although same data are input to both systems, each system processes some of them because of the difference of the processing speed.

4. The classification ratios of our method, the simple method and the previous method are calculated by each output.

In this experiment, the classification ratio is the F measure (F_{ob}) calculated by the recall ratio R_{ob} and the precision ratio P_{od} as shown in Equation 7.4, 7.5 and 7.6. Here, A_{ob} , O_{ob} and C_{ob} show the number of images to which another moving robot is projected, the number of obstacles the system classified as dynamic obstacles, and the number of dynamic obstacles the system outputs and locates correctly, respectively.

$$R_{ob} = \frac{C_{ob}}{A_{ob}} \tag{7.4}$$

$$P_{ob} = \frac{C_{ob}}{O_{ob}} \tag{7.5}$$

$$F_{ob} = \frac{2P_{ob}R_{ob}}{P_{ob} + R_{ob}} \tag{7.6}$$

7.3.2 Results

The classification ratios of both methods are shown in Table 7.2. The classification ratio of our method is about 4 times higher than that of the simple method and the previous method. In particular, the improvement of the precision ratio affects the F measure.

Table 7.2: The classification ratios

Method	Recall Ratio	Precision Ratio	F measure
Previous	0.18 (3/17)	0.25 (3/12)	0.21
Simple	0.63 (10/16)	0.13 (10/79)	0.21
Dynamic	0.94 (17/18)	0.77 (17/22)	0.85

7.4 Confirmation of Ability of Obstacle Classification by ApriTau^{TM}

7.4.1 Aim and Sequence of Experiment

In order to confirm our method detects moving people, we calculate the classification ratio in various patterns. In this experiment, a person and ApriTau^{TM} move on the given route as shown in Figures 7.10, 7.11, and 7.12. In order to confirm basic ability of our method, ApriTau^{TM} and one person go straight and rotate. As same as the experiment in Section

7.3, ApriTau^{TM} takes images synchronized with odometry data and the classification ratio of our method is calculated.

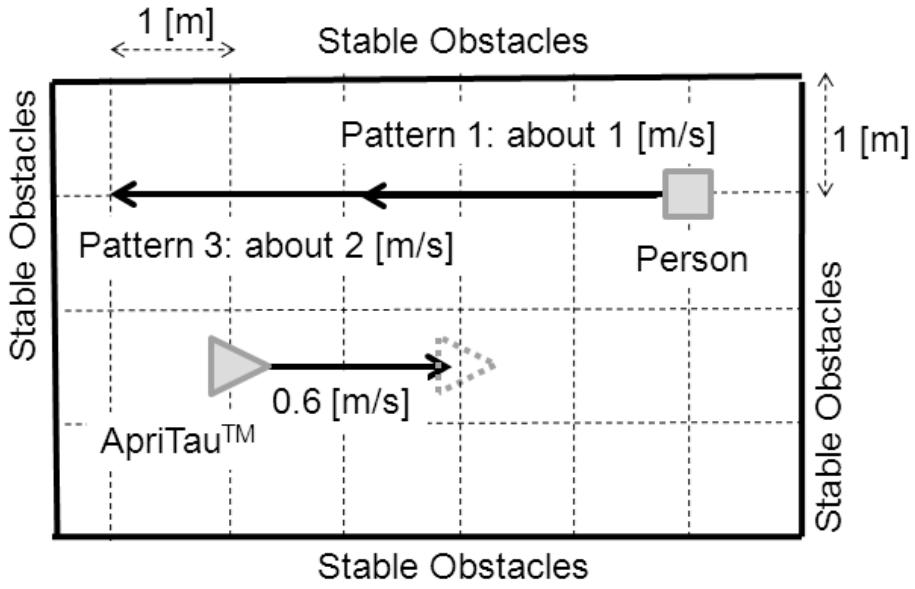


Figure 7.10: The experimental setting (Pattern 1 and 3) (5, Figure 12)

7.4.2 Results

The classification ratios in each pattern are shown in Table 7.3. The distance shownin the Table denotes the distance where the robot first detects a person. The classification ratios in the case of the person walking (Patterns 1 and 2) are higher than 0.77, which is as high as the classification ratios in Section 7.3. The classification ratios in the case of the person running (Patterns 3 and 4) are a little low. One of the reasons why the classification ratios are a little low is that the boundary between the running person and the floor is more complex than the boundary between the walking person and the floor. The complex boundary can make robots fail to detect floor boundary points accurately. We think that increasing floor boundary points can solve this problem.

The classification ratio in the case of the robot rotation (Pattern 5) is also a little low. One of the reasons why the classification ratio is a little low is that tracking area in the image in the case of rotation changes more than tracking area in the case of straight transition (Patterns 1—4) does. Changing tracking area very much makes robots fail to track the floor boundary points. Moreover, we need to synchronize the timestamps

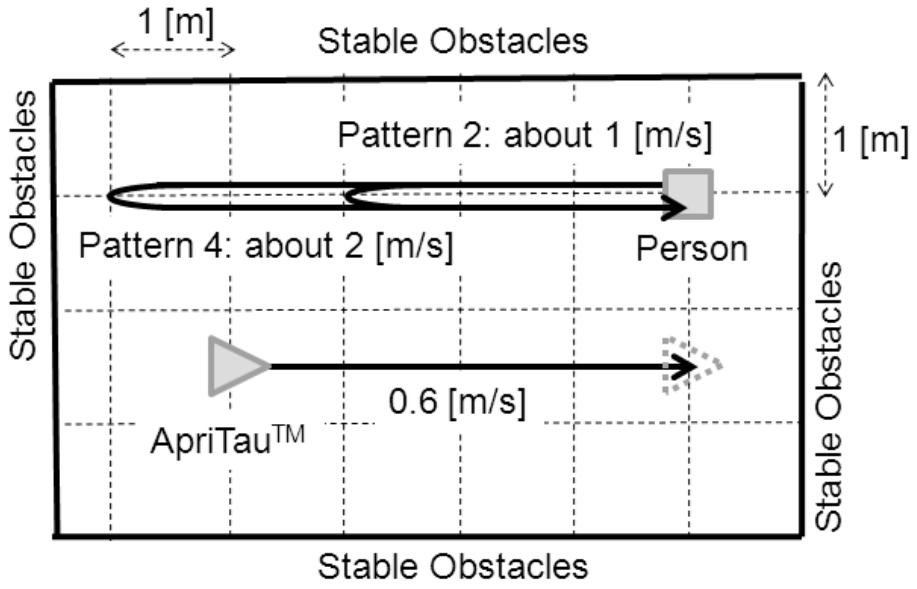


Figure 7.11: The experimental setting (Pattern 2 and 4) (5, Figure 13)

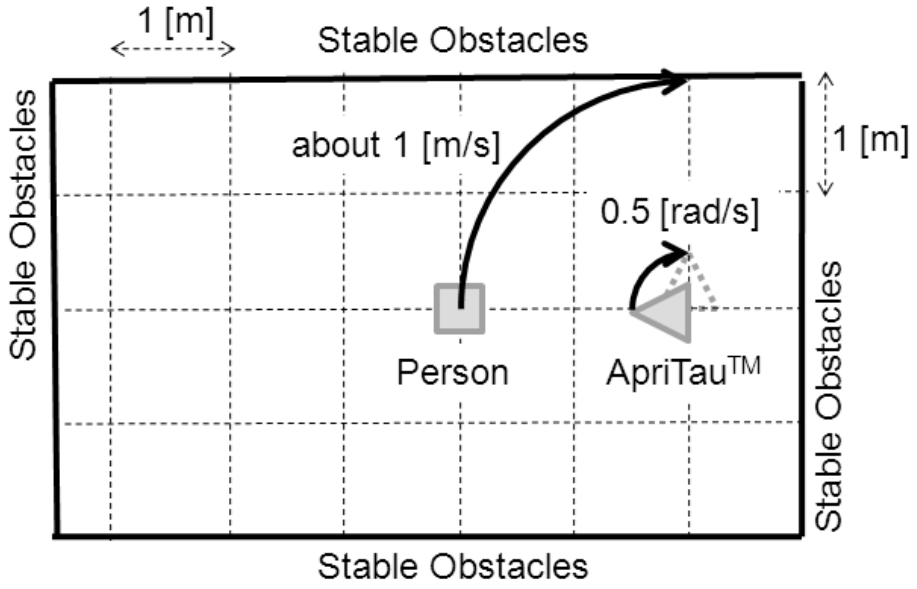


Figure 7.12: The experimental setting (Pattern 5) (5, Figure 14)

Table 7.3: The classification ratios in various cases

Pattern	Recall Ratio	Precision Ratio	F measure	Distance [m]
1	1.00 (21/21)	0.64 (21/33)	0.79	2.7
2	0.98 (42/43)	0.64 (42/67)	0.77	2.8
3	0.80 (16/20)	0.64 (16/29)	0.71	3.1
4	0.93 (42/45)	0.59 (42/74)	0.72	2.9
5	0.93 (13/14)	0.57 (13/23)	0.71	2.0

between odometry and images. We also think that it is effective to take into account uncertainty in sensing. The accuracy of odometry or tracking differs according to the robot movement. We have to use probabilistic method in the future work.

7.5 Evaluation of Landmark Selection by ApriTau^{TM}

7.5.1 Aim and Sequence of Experiment

In order to confirm a validity of the feature vector and decide the appropriate parameters, we evaluate the landmark selection from among many tracked LI feature points by the SVM. The experimental steps are as follows:

1. We localize landmark condidates (tracked LI feature points) by the least square method after ApriTau^{TM} moves along the route shown in Figure 7.13.

2. The 100 landmark condidates are selected randomly.

3. The landmarks condidates are classified as correct location or not by the SVM.

4. In the previous step, we calculate classification ratio, changing the feature vector and the parameters of the SVM.

In the last experimental step, we use the 6 dimentional feature vector as shown in Chapter 6 and 5-dimentional feature vectors which are made by removing 1 feature from the 6-dimentional feature vector. Here, the classification ratio R_{mark} is defined by Equation

7.7. N_{mark} denotes the number of the landmarks located correctly and N_{all} denotes the number of landmarks classified "correct" by SVM.

$$R_{mark} = 100 \times \frac{N_{mark}}{N_{all}} \tag{7.7}$$

We evaluate the classification ratio by the cross validation. For the cross validation, the 100 landmarks are divided into 5 groups. Each group has 20 landmarks. In the 100 landmarks selected randomly, 39 landmarks are located correctly. Therefore, the selection method by the SVM is effective when the classification ratio is more than 39 [%].

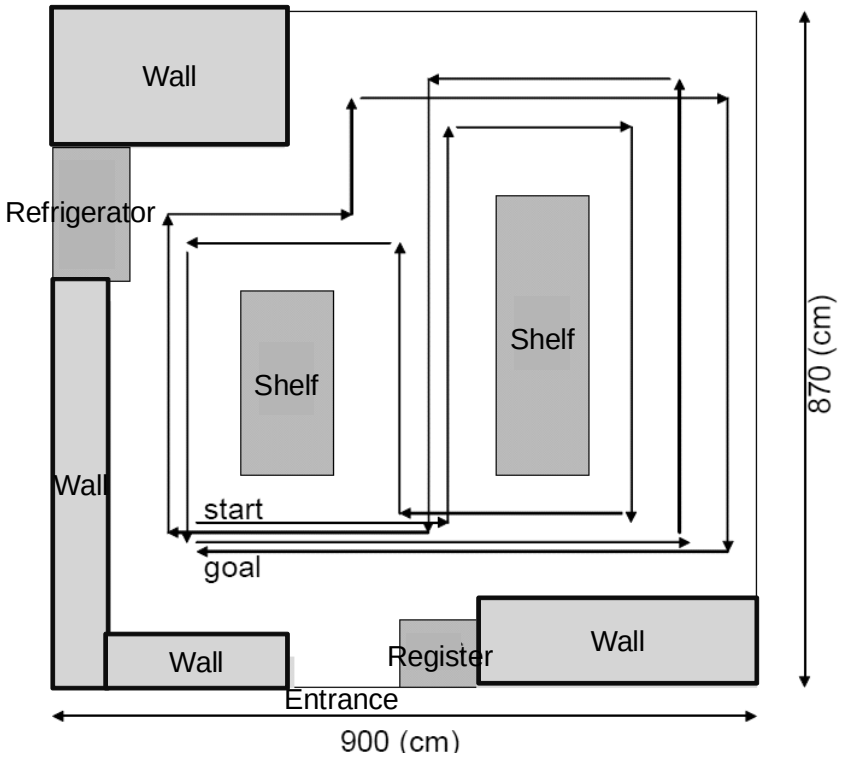


Figure 7.13: The route of ApriTau^{TM} for the landmark selection

7.5.2 Results

Figure 7.14 shows the relationship between the parameter value and the classification ratio of each feature vector. The horizontal axis in the Figure 7.14 shows the parameter value of SVM which denotes the coefficient $1/(2\sigma^2)$ described in Equation 7.1.

Figure 7.14 shows that the SVM performs the best when it uses 5 dimentional feature vector and its parameter denotes 0.6. The 5-dimentional feature vector consists of 5

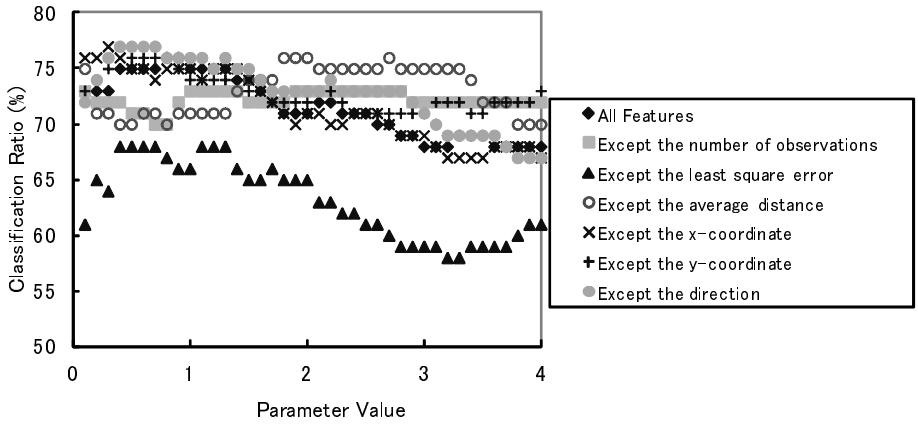


Figure 7.14: The relationship between the parameter value and the classification ratio

feature, the number of observations, the least square error of the measurement [rad^2], the average distance from the robot to the points in the image [pixel], the distribution of the observation positions' x-coordinate [m^2] and the distribution of the observation positions' y-coordinate [m^2].

Table 7.4 shows the confusion matrix when the SVM performs the best. The SVM classifies 77 (= 29 + 48) landmarks correctly. The 42 (= 29 + 13) landmarks are classified as the correct location by the SVM and the classification ratio is 69.0 [%] = $(100 \times 29/42)$. The classification ratio is 1.8 times higher than 39 [%]. Therefore, the SVM is effective.

Table 7.4: The confusion matrix

	Success	Fail
Success	29	10
Fail	13	48

7.6 Accuracy and Computational Time Evaluation of Self-Localization by ApriTau^{TM}

7.6.1 Aim and Sequence of Experiment

In order to confirm the effectiveness of the tracked LI feature points, we compare two methods. One is our developed method that switches between the SURF method and the

tracking method. The other is a simple method that uses only the SURF method. The experimental steps are as follows:

1. ApriTau^{TM} moves along whole paths at the experimental room and selects landmarks.

2. ApriTau^{TM} moves along a fixed route for evaluation. In this case, we use a 4 [m] line by a wall for simplicity. ApriTau^{TM} takes continuous omnidirectional images while moving.

3. Both our method and the simple method estimate the positions of ApriTau^{TM} at each time and measure the computational time of processing one omnidirectional image.

4. We use the Euclidean distance between the estimated positions and the route at each time as the error. For evaluation, we use an average (avg.) and a standard deviation (SD) of the error calculated by each method. We also use an avg. and a SD of the computational time. We compare our method with the simple method based on these 4 values.

7.6.2 Results

Figure 7.15 shows the estimated position of both our method and the simple method. A line by the y-coordinate 2 [m] shows the route of ApriTau^{TM}. Squares and triangles denote the estimated positions of our method and the simple method, respectively.

Table 7.5 shows the avg. and SD of the error and the avg. and SD of the computational time calculated by both methods. As shown in Table 7.5, the error of our method is reduced to 23.6% of the simple method. The computational time of our method is also 2.9 times faster than that of the simple method. The computational time of our method is much shorter than that of the simple method, which is confirmed by t-test.

Table 7.5: Comparing our method with the simple method in terms of the localization errors and computational time

Method	Error Avg.	Error S. D.	Time Avg.	Time S. D.
Ours	0.38 [m]	0.21 [m]	16.4 [ms]	4.65 [ms]
Simple	1.61 [m]	1.36 [m]	46.8 [ms]	1.81 [ms]

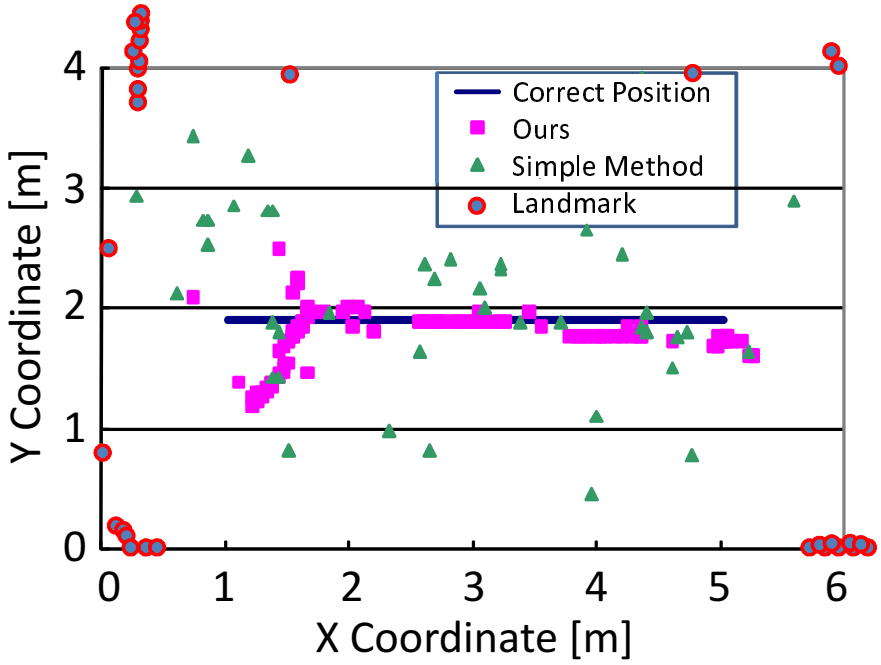


Figure 7.15: The localization results of the developed method and the simple method (7, Figure 11)

7.7 Summary

In this chapter, we evaluate our moving people detection and fast self-localization method. For evaluation, we use AprilTau^{TM} at a mock shopping store whose size is 10 [m] $\times$ 10 [m]. An omnidirectional camera is mounted on the top of its head. It moves at 0.6 [m/s]. Our method can detect dynamic obstacles 4.0 times more accurately than the previous method can. When we use moving people as dynamic obstacles, AprilTau^{TM} detects 74% of moving people. Moreover, our method enables AprilTau^{TM} to localize itself 2.9 times faster than the previous method does. The error of localization is reduced to 23.6% of the error of the previous localization.

Chapter 8

Discussion

This chapter first demonstrates the service for visitors in the mock shopping store and second discusses the main contributions of this study, then discusses the remaining issues and future work.

8.1 Demonstration of Navigation and Transportation (Proof of Concept)

Before operating the robot system for navigation and transportation at actual shopping store, we demonstrate a sequence of the services. Visitors go shopping and our robots provide services as follows:

1. The visitor pushes the button on the touch panel and registers his/her cloth's textures to the robot.

2. If the visitor needs the navigation service, he/she pushes the position on a map displayed on the touch panel. Then, the robot moves to the place corresponding to the position.

3. If the visitor does not need the navigation service, the robot follows him/her by using the registered information.

4. When the visitor asks the robot to transport his/her baggage, he/ser pushes a transportation service button. The robot serves a shopping cart and puts it in front of him/her.

5. When the visitor finishes shopping, he/she pushes an end button. The robot transports his/her selected commodities to a self-checkout.

Figures 8.1 and 8.2 show the examples of shopping with ApriTau^{TM}.

Here, we use a cart robot with ApriTau^{TM}, too. We developed two types of robots because it is unclear that a humanoid type robot is the best for the low-cost mobile robots at the big shopping store. We have to confirm the appropriate types of the robot.

8.2 Contributions

In this thesis, we addressed three issues in development of service robots at the big shopping store, (1) detecting the interaction partner, (2) detecting moving people while robot moves by a single omnidirectional camera, and (3) detecting landmarks fast by the omnidirectional camera.

Regarding to the interaction partner detection, there are two problems: integration of multiple sensors in the noisy environments and selection of the interaction partner from among multiple people. To solve these two problems, we used the interaction distance to integrate multiple sensors and to select the interaction partner from among multiple people.

To solve the second issue, we regard te obstacles as floor boundary points. The local point is not affected by the distortion of the omnidirectional image. Tracking them, we classify the point whose movement is different from the movement of the robot as a dynamic obstacle.

To solve the third issue, we use the continuity of landmarks. We regard the SURF feature points which can be tracked by the fast tracking method as landmarks. Because the landmarks can be tracked, the robot can detect them fast.

The main contributions of these are summarized as follows:

Towards robot and multiple people interaction

We can verify that someone is in the space with the highest friendliness level, since the friendliness space map considers the human existence degree. For discussions, we point out the results presented in Chapter 7 showed that the recall ratio was low. This is because the people did not use SIG2's functions positively to interact between the person and the person ignoring SIG2. This is a special problem for interaction between a robot and "multiple" people. Therefore, it is interesting to develop the method which enables a robot to join the interaction between multiple people appropriately.

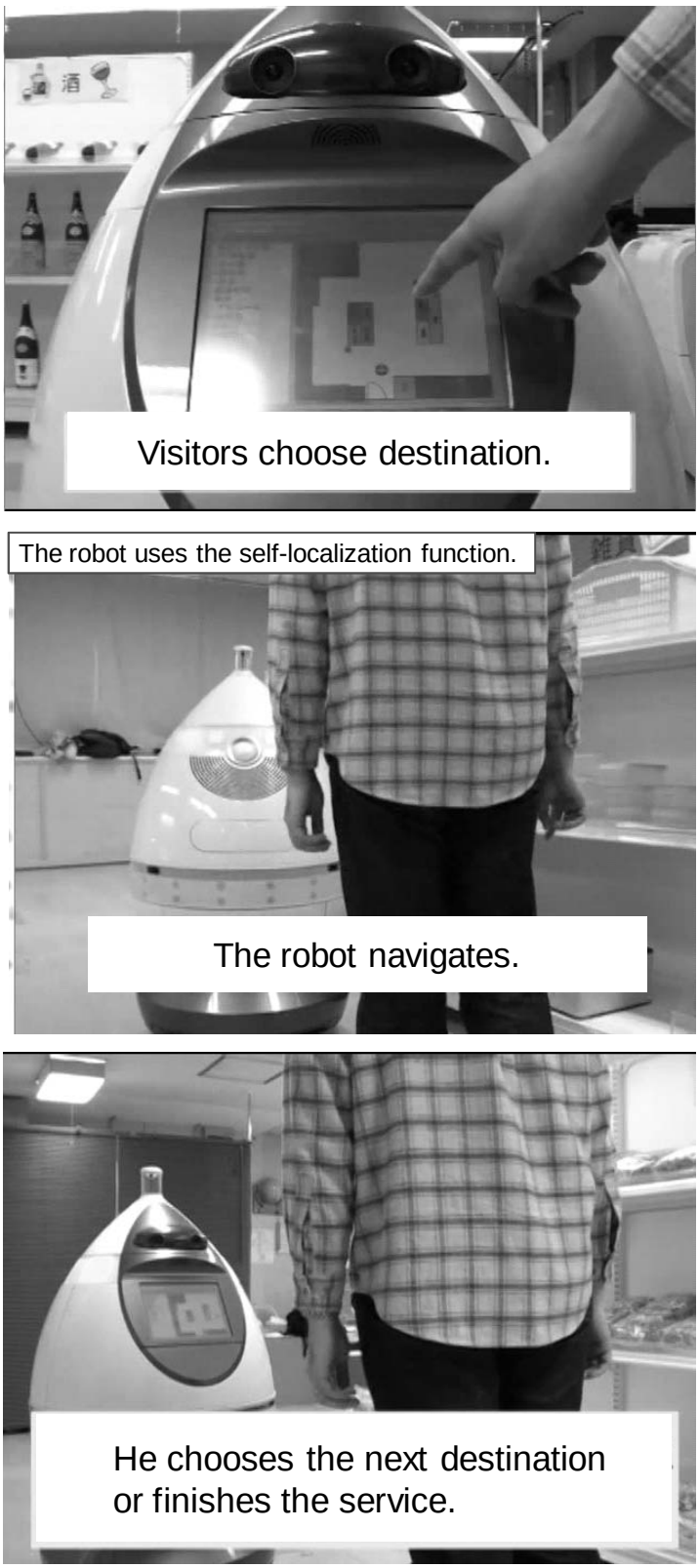


Figure 8.1: The demonstration of the navigation service

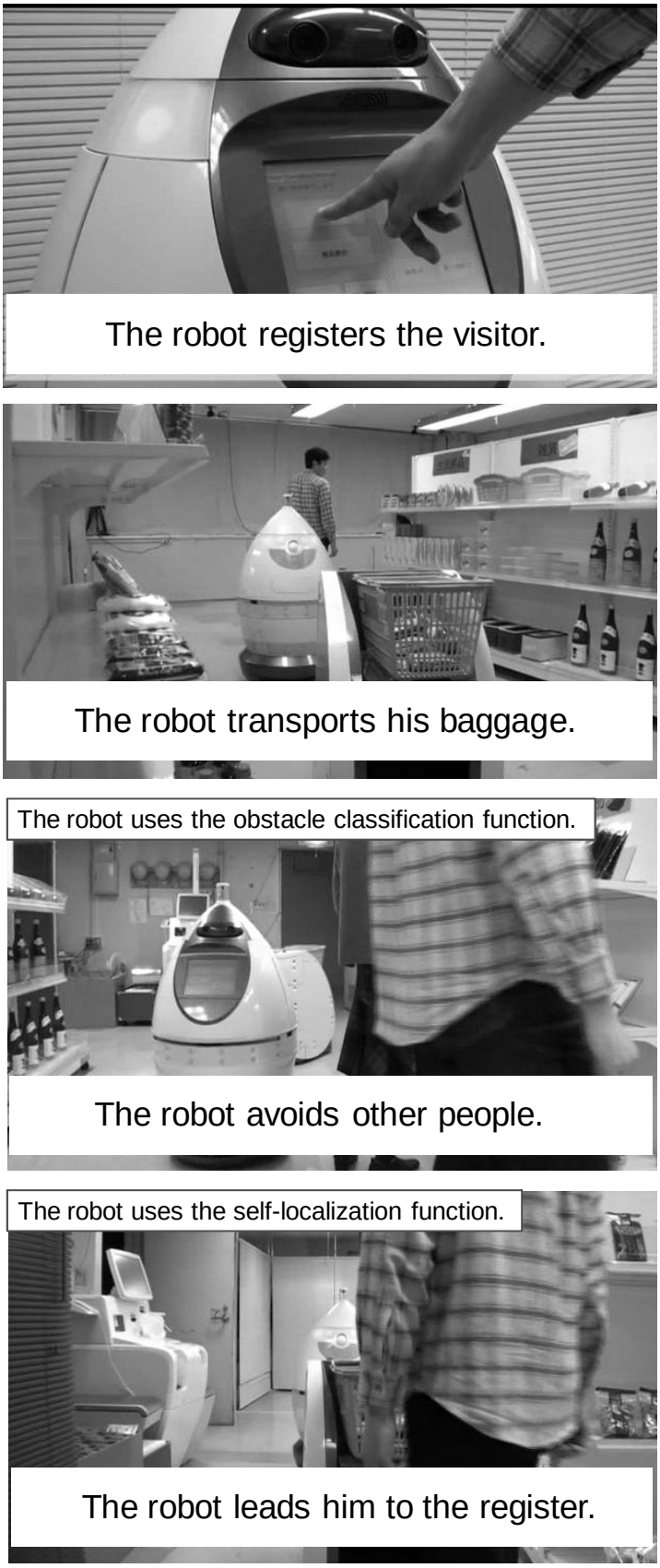


Figure 8.2: The demonstration of the transportation service

We can also verify that the friendliness space map makes SIG2 behave friendly, when it interacts with multiple people. For the times when SIG2 behaved based on the friendliness space map, the impression scores for the adjectives related to the first factor were high. This is because the simple selection criteria based on the friendliness made SIG2 behaviors seem plausible. For the times when the map was not used, the simple behaviors caused lower impression scores. In particular, because the score of the second factor including "Friendly" was higher, we think the definition of friendliness based on the distance fits the friendliness that people feel. Since the "Exciting" impression for the interaction with the robot was strong, the factor including "Friendly" was second. Moreover, in order to discuss the relationship between our defined friendliness and the results of questionnaires, we use the average of our defined continuous friendliness. The average is 0.24, which is positive. Therefore, we think our defined friendliness may relate to the positive impressions obtained by questionnaires. However, the standard deviation is 1.1. That is, our friendliness becomes sometimes negative. The reason why the friendliness is sometimes negative is because there are some people who hit the robot as an interaction. We design hitting is uncomfortable stimuli, but that depends on the context. In future work, we have to consider context to design the comfortable degree.

Towards detecting moving people by a mobile omnidirectional camera

Our method first detects floor region and classify obstaces as stable or not. Our method detects the floor region by the Ward's method. Our method works better than the previous GMM method does. One of the reasons why the second Hit value calculated by the result of the GMM in Chapter 7 is low is that many experiments are needed in order to decide the parameter of the GMM. Comparing the GMM parameters, parameters of the Ward's clustering and the threshold of Mahalanobis distance do not change dramatically because they are related to the distribution of learning data and determined based on them. In our experiment, the second test data includes more colors than the first test data does because of illumination changing and so on. The number of mixture Gaussian optimized at the first test is too small for the GMM to learn the floor colors completely.

Next contribution is that we can clasify all obstacles around the robot by a single mobile omnidirectional camera, which was difficult previously. One of the reasons why the precision ratio of our method is much higher than that of the simple method is that

ApriTau^{TM} can select floor boundary points which are candidates of dynamic obstacles by the CE and relocate them correctly by strengthening the threshold to detect each point. The result shows obstacles can be classified even if we regard obstacles as small points. It also shows that the accuracy of locating points greatly affects classification ratio.

For discussions, we point out that the precision ratio is a little low. In this thesis, we assume errors of tracking points are very small, which is certainly correct to some extent for the image coordinates. In the case of omnidirectional camera image, the distance resolution changes depending on the distance from the image center. It is very low for a distant place. Tracking errors of a few pixels become errors of a few meters for the world coordinates. Because of errors of a few meters, Equation 5.9 does not work as the CE. When the floor boundary point is located at a position distant from the center of the image, we may track it for a longer time and use its average movement.

The average of F measure is 0.75. One of the reasons why our robot cannot detect people is because our method cannot work when the distance between the robot and the objects is long. The average of distance where our robot first detects the person is 2.9 [m]. In these experiments, the robot starts to move at 4.0 [m] away from the person. Therefore, our robot did not detect the person while the distance is more than about 3.0 [m]. Our robot moves by 0.6 [m/s]. The people generally walk by 1.0 [m/s]. We think 2.9 [m] is enough long to avoid people, considering their velocities. However, when the people run by more than 1.0 [m/s], we have to make the robot move slowly or stop.

In the assumed shopping stores, we found that it was difficult for our method to detect thin legs of shelves because of the low resolutions. However, the shelf boards are detected. Using the information our robot can find the boards, our robot moves about 50 [cm] away from the detected boundary in order to avoid legs.

Moreover, it is also difficult to classify obstacles as stable or not when there are a lot of people around the robot. However, our robot can recognize there are a lot of people around the robot. For safety, our robot does not move in that case. In order to provide services, we have to integrate planning methods with our methods, considering the accuracy of our methods.

Towards self-localization by an omnidirectional camera

Our method contributes the fast self-localization by an omnidirectional camera. The computational time of our method is much shorter than that of the simple method. The

integration of the fast tracking method is effective. In order to discuss limitations, we point out that the computational time standard deviation (S.D.) of our method is longer than that of the simple method. We think the S.D. is long because the computational time of "Sub Flow" is much longer than that of "Main Flow". In a real shopping store, there are many people. Therefore, we think our system often switches between the SURF method and the tracking method, and our method performs slowly. It is interesting to confirm how fast our method can perform in the crowded area.

Our method contributes more accurate self-localization by an omnidirectional camera. The robot uses tracked LI feature points that can be detected at various positions and have salient feature vectors. In order to discuss limitations, as shown in Figure 7.15, our method cannot work accurately from 1.0 [m] to 1.5[m] on x-coordinate. This error occurs because a white wall without landmarks exists on an x-coordinate 0 [m] line. If the robot uses only a general camera that has a narrow field of view, it cannot localize its position at all in front of the white wall. On the other hand, ApriTau^{TM} can localize fast with about 60 [cm] errors even in this case, which is an advantage. For example, using the surveillance cameras that have been already installed, the robot can localize its position very accurately.

Towards safety of service robots

We developed a navigation and transportation system. In order to apply our system to the real shopping stores, verification of safety is one of the most important issues. For safety, obstacle classification function is important. In the system, our method can detect 74% of moving people. Our method cannot detect all moving people because of low resolution of the omnidirectional images. Specially, the resolution is very low at far place. We performed additional experiments which showed that our method detected people who move at 2.9 [m] away from the robot. Considering to process time of our obstacle classification (30 [fps]), the velocity of the robot (0.6 [m/s]) and the average velocity of the people (1.0 [m/s]), the robot is close to people by only 5.3 [cm] during processing. Therefore, our method can make it possible to avoid people unless people suddenly stand up in front of the robot.

8.3 Remaining Issues and Future Work

In this section, we discuss remaining issues for practical applications of our service robot in the shopping store and future work.

- **Long term interaction**

  One of the reason why selecting the interaction partner by friendliness makes people feel friendly is because the friendliness holds a simple short interaction history. Long interaction history may make people more plausible in the long term interaction. In future work, we have to develop history management system which holds important history for the interaction based on values such as friendliness.

  In this work, the behavior of SIG2 with interaction partners is simple. Therefore, if multiple people interact with SIG2 more than a few minutes, the person-to-person interactions increase, and SIG2 may lose its impression scores. For more active interaction, the robots must interact appropriately to impress the people. We plan to implement the proposed method in a robot that has many degrees of freedom and behaves using Q-learning with friendliness as a reward.

- **Path planning under the crowded condition**

  In this work, the omnidirectional camera enables the robot to detect all obstacles around the robot and classify them as stable or not. Moreover, our method works fast enough to control the robot to avoid obstacles. However, it cannot move under the crowded condition even if it detects all obstacles. In the case of people, they can walk with contacting each other. In the future work, we need path planning with contacting each other based on the information of obstacles.

- **Changing location of the landmarks**

  We select landmarks which are detected fast for a long time based on the constraint of continuity. Some selected landmarks located at high positions such as ceiling lights and signs of commodities do not change their locations in the short term less than a month. Other selected landmarks are commodities which have rich textures. The locations of them change frequently. The change of the landmark location decreases the accuracy of self-localization. In future work, we have to remove the selected landmarks of which the locations may change such as the landmarks located at low positions.

- **Cooperation with environmenatal sensors and other robots**

  We realize fast obstacle classification and self-localization by using a single omni-directional camera. We use only one sensor, considering the cost. There are no problems we can use sensors which the shopping store has already installed. One of the most effective sensors is a surveillance camera. The surveillance cameras are useful to detect obstacles which are not observed from the robot and to make it possible to localize the robot more accurately.

  We can also use other robot's information because we made a lot of robots in order to provide services to a lot of visitors. Robots can communicate the information of people location with other robots. Robots also can observe corner in order to avoid collision when the robot provides no services. Communicating information make it possible to avoid people who suddenly stand up in front of robots.

  In this work, we make the robot move slowly when the robot turns a corner. Cooperation with environmental sensors and other robots will enable our robot to move at a corner smoothly.

Chapter 9

Conclusion

In this thesis, we dealt with the detection of people and objects for a robot service in big shopping stores. We pointed out three issues to achieve such a robot system as follows:

Issue 1 Detecting an interaction partner based on the design of plausible relationship to interact with multiple people,

Issue 2 Classifying all obstacles around the robot as stable or not by a single omnidirectional camera while robot moves,

Issue 3 Fast and accurate landmark detection by the omnidirectional camera.

In order to solve the first problem, we focus on the distance between the interaction robot and each person. For robots interacting with multiple people, the space around the robot is divided based on the interaction distance and the effective distances of multiple sensors for interaction. We have developed a method for selecting an interaction partner based on the degree of friendliness as mapped onto the "space".

In order to solve the second problem, we focus on small feature points (floor boundary points) which are not affected by the distortion of the omnidirectional images. We have developed a new method that focuses on floor boundary points where the robot can measure the distance from itself by a single omnidirectional camera. Our robot classifies a floor boundary point as a moving person when its movement is different from the robot's movement.

In order to solve the third problem, we focus on the continuity of SURF feature points. We have developed a new method that uses tracked local invariant feature points that can be detected by both fast tracking method and slow SURF method. Once we detect the landmarks by slow SURF, we detect them by fast tracking method.

We summarize each chapter as follows.

Chapter 3 describes our friendliness space map. We show the design of the map based on the interaction distance. And we show the design of mapping friendliness based on the sensor input in order to select an interaction partner among multiple people.

In Chapter 4, we show the interaction robot SIG2. We show that the interaction based on the friendliness makes multiple people feel friendly.

Chapter 5 describes the obstacle classification method based on tracking floor boundary points by using a single omnidirectional camera while a robot moves. We find the points based on the Ward's clustering and track them with changing thresholds dynamically based on the results of the classification.

Chapter 6 describes how we detect tracked local invariant feature points by an omnidirectional camera. We regard the points which can be detected by both SURF and the tracking method as landmarks. We explain how to use them for the fast self-localization.

In Chapter 7, we show the classification ratio of our method is 4.0 times higher than that of the previous method. We also show our self-localization works 2.9 times faster than the previous method does. Moreover, our method can reduce the error of localization to 23.6% of the previous method.

In Chapter 8, we discuss the major contribution of this work towards the robot system at a big shopping store. We describe remaining issues and future directions of our work.

We thus achieved development of the method detecting people and objects for service robots that can interact with multiple people simultaneously and navigate them accurately. We hope that our study will trigger further attempts to develop a robot service system that works with people.

Relevant Publications

Chapter 3
Interaction Partner Detection by Spatial Mapping of Friendliness Based on Interaction Distance

(1) Tsuyoshi Tasaki, Shohei Matsumoto, Hayato Ohba, Shunichi Yamamoto, Mitsuhiko Toda, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno: Dynamic Communication of Humanoid Robot with Multiple People Based on Interaction Distance, Transactions of the Japanese Society for Artificial Intelligence, Vol.20, No.3, pp.209-219, 2005.

(2) Tsuyoshi Tasaki, Shohei Matsumoto, Hayato Ohba, Mitsuhiko Toda, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno: Distance Based Dynamic Interaction of Humanoid Robot with Multiple People, Innovations in Applied Artificial Intelligence, pp. 111-120, 2005. (**Best paper award**)

Chapter 4
Implementation and Evaluation of Detecting People for Interaction Robot

(3) Tsuyoshi Tasaki, Tetsuya Ogata, Hiroshi G. Okuno: The Interaction between a Robot and Multiple People based on Spatially Mapping of Friendliness and Motion Parameters, Advanced Robotics (accepted), 2013.

(4) Tsuyoshi Tasaki, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno: Spatially Mapping of Friendliness for Human-Robot Interaction, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1277-1282, 2005.

Chapter 5
Obstacle Classification and Location by Using a Mobile Omnidirectional Camera Based on Tracked Floor Boundary Points

(5) Tsuyoshi Tasaki, Seiji Tokura, Takafumi Sonoura, Fumio Ozaki, Nobuto Matsuhira: Classification of Static and Dynamic Obstacles Based on Tracked Floor Boundary Points by an Omnidirectional Camera Mounted on a Mobile Robot Journal of the Robotics Society of Japan, Vol.29, No.5, pp.68–77, 2011.

(6) Tsuyoshi Tasaki, Fumio Ozaki: Obstacle classification and location by using a mobile omnidirectional camera based on tracked floor boundary points, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5222-5227, 2009.

Chapter 6
Mobile Robot Self-Localization Based on Tracked Local Invariant Feature Points by Using an Omnidirectional Camera

(7) Tsuyoshi Tasaki, Seiji Tokura, Takafumi Sonoura, Fumio Ozaki, Nobuto Matsuhira: Obstacle Location Classification and Self-Localization by Using a Mobile Omnidirectional Camera Based on Tracked Floor Boundary Points and Tracked Scale-Rotation Invariant Feature Points, Journal of Robotics and Mechatronics, Vol.23, No.6, pp.1012–1023, 2011.

(8) Tsuyoshi Tasaki, Seiji Tokura, Takafumi Sonoura, Fumio Ozaki, Nobuto Matsuhira: Mobile robot self-localization based on tracked scale and rotation invariant feature points by using an omnidirectional camera, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5202-5207, 2010.

List of All Publications by Author

International Conference

(1) Tsuyoshi Tasaki, Akihisa Moriya, Aira Hotta, Takashi Sasaki, Haruhiko Okumura: Depth Perception Control by Hiding Displayed Images Based on Car Vibration for Monocular Head-up Display, IEEE International Symposium on Mixed and Augmented Reality Science and Technology Proceedigs, pp. 323-324, 2012.

(2) Tsuyoshi Tasaki, Takeshi Yamaguchi, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno: Robot Motion Control using Listener's Back-Channels and Head Gesture Information. Proceedings of International Conference on Spoken Language Processing, pp. 1033-1036, 2004.

Other Publications (All in Japanese)

(3) Tsuyoshi Tasaki, Daisuke Yamamoto, Miwako Doi: Detecting Elderly People Getting-up by Cameras Using Parameters Setting Based on a Distance between a Robot and a Person. Proceedings of the Annual Conference of the Human Interface Society of Japan, 2011.

(4) Tsuyoshi Tasaki, Daisuke Yamamoto, Miwako Doi: Mapping a Room by a Watching over Robot and an Application to an Exit Alarm. Proceedings of the 26th Fasy System Symposium, 2010.

(5) Tsuyoshi Tasaki, Fumio Ozaki: Obstacle Classification by a Mobile Omni-Directional Cameras Based on Tracked Floor Boundary Points, Proceedings of the 26th Annual Conference of the Robotics Society of Japan, 2008.

(6) Tsuyoshi Tasaki, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno: Robot Motion Selection Based on the Position of People by Using Reinforcement Learning,

Proceedings of the 68th IPSJ National Convention, 2006.

(7) Tsuyoshi Tasaki, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno: Spatial Mapping of Friendliness for Robot Interaction with Multiple People, Proceedings of the 23th Annual Conference of the Robotics Society of Japan, 2005.

(8) Tsuyoshi Tasaki, Shohei Matsumoto, Hayato Ohba, Masamitsu Murase, Taku Ohya, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno: Spatial Mapping of Friendliness with People by Humanoid and Its Application to Humanoid Human Interaction, Proceedings of the 67th IPSJ National Convention, 2005.

(9) Tsuyoshi Tasaki, Shohei Matsumoto, Hayato Ohba, Mitsuhiko Toda, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno: Proxemics-based Interaction of Humanoid SIG2 with People, Proceedings of the 22th Annual Conference of the Robotics Society of Japan, 2004.

(10) Tsuyoshi Tasaki, Takeshi Yamaguchi, Mitsuhiko Toda, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno: Backchannels recognition based on multimodal information and its application to robot dialogue, Proceedings of the 66th IPSJ National Convention, 2004.

Bibliography

[1] Shunichi Yamamoto, Kazuhiro Nakadai, Mikio Nakano, Hiroshi Tsujino, Jean-Marc Valin, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Simultaneous Speech Recognition Based on Automatic Missing Feature Mask Generation by Integrating Sound Source Separation. *Journal of The Robotics Society of Japan*, 25(1):92–102, 2007.

[2] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The Intelligent ASIMO: System overview and integration. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 2478–2483, 2002.

[3] Kazumi Aoyama and Hideki Shimomura . Real world speech interaction with a humanoid robot on a layered robot behavior control architecture. In *Proceedings of International Conference on Robotics and Automation*, pages 3814–3819, 2006.

[4] Junko Hirokawa, Nobuto Matsuhira, Hideki Ogawa, and Tatsuya Wada. Universal Design with Robos: Practice of House-working Task of Home Robot. *Journal of The Robotics Society of Japan*, 26(6):476–484, 2008.

[5] Takashi Yoshimi, Manabu Nishiyama, Takafumi Sonoura, Hideichi Nakamoto, Seiji Tokura, Hirokazu Sato, Fumio Ozaki, Nobuto Matsuhira, and Hiroshi Mizoguchi. Development of a Person Following Robot with Vision Based Target Detection. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 5286–5291, 2006.

[6] Noriaki Mitsunaga, Zenta Miyashita, Takahiro Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. Robovie-IV : An Everyday Communication Robot and its Daily Communication in an Office. *Journal of The Robotics Society of Japan*, 25(8):1243–1250, 2007.

[7] Syuichi Nishio, Norihiro Hagita, Takahiro Miyashita, Takayuki Kanda, Noriaki Mitsunaga, Masahiro Shiomi, and Tatsuya Yamazaki. Robotic Platforms Structuring Information on People and Environment. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 2637–2642, 2008.

[8] Dylan F. Glas, Takahiro Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. Laser Tracking of Human Body Motion Using Adaptive Shape Modeling. *Journal of Advanced Robotics*, 23(4):405–428, 2009.

[9] Shinji Kanda, Yuichi Murase, Naoyuki Sawazaki, and Tsutomu Asada. Development of Service Robot enon and Its Applications. *Journal of The Robotics Society of Japan*, 24(3):12–15, 2006.

[10] Zenta Miyashita, Takayuki Kanda, Masahiro Shiomi, Hiroshi Ishiguro, and Norihiro Hagita. A Robot in a Shopping Mall that Affectively Guide Customers. *Journal of The Robotics Society of Japan*, 26(7):821–832, 2008.

[11] Ryosuke Murai, Tatsuo Sakai, Hiroyuki Uematsu, Hisato Nakajima, Koichi Mitani, and Hitoshi Kitano. Practical Design and Use of Transfer System by Autonomous Mobile Robot Group. *Journal of The Robotics Society of Japan*, 28(3):311–318, 2010.

[12] Masaki Takahashi, Takafumi Suzuki, Toshiki Moriguchi, Hideo Shitamoto, and Kazuo Yoshida. Developing a mobile robot for transport applications in the hospital domain. *Journal of Robotics and Autonomous Systems Advances in Autonomous Robots for Service and Entertainment*, 58(7):889–899, 2007.

[13] Heesung Chae, Christiand, Sunglok Choi, Wonpil Yu, and Jaeil Cho. Autonomous navigation of mobile robot based on DGPS/INS sensor fusion by EKF in semi-outdoor structured environment. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 1222–1227, 2010.

[14] Shinya Fujie, Yasushi Ejiri, Kei Nakajima, Yosuke Matsusaka, and Tetsunori Kobayashi. A Conversation Robot Using Head Gesture Recognition as Para-Linguistic Information. In *Proceedings of International Workshop on Robot and Human Communication*, pages 159–164, 2004.

[15] Hiroshi G. Okuno, Kazuhiro Nakadai, Ken ichi Hidai, Hiroshi Mizoguchi, and Hiroaki Kitano. Human-Robot Non-Verbal Interaction Empowered by Real-Time Auditory

and Visual Multiple-Talker Tracking. *Journal of Advanced Robotics*, 17(2):115–130, 2003.

[16] Frederic Kaplan and Verena V. Hafner. The Challenge of Joint Attention. In *Proceedings of International Workshop on Epigenetic Robotics*, pages 67–74, 2004.

[17] Cynthia Breazeal and Lijin Aryananda. Recognition of Affective Communicative Intent in Robot-Directed Speech. *Journal of Autonomous Robots*, 12(1):83–104, 2002.

[18] Tetsuya Ogata and Shigeki Sugano. Mechanisms of Internal Secretion System for Intellectual Robots -Towards an Emergence of Emotion in Robots. In *Proceedings of International Workshop on Robot and Human Communication*, pages 50–55, 1998.

[19] Frederic Kaplan and Verena Hafner. A New Mental Model for Humanoid Robots for Human Friendly Communication -Introduction of Learning System, Mood Vector and Second Order Equations of Emotion-. In *Proceedings of International Conference on Robotics and Automation*, pages 3588–3593, 2003.

[20] Takuya Murakita, Tetsushi Ikeda, and Hiroshi Ishiguro. Human Tracking using Floor Sensors based on the Markov Chain Monte Carlo Method. In *Proceedings of International Conference on Pattern Recognition*, pages 917–920, 2004.

[21] Takayuki Kanda and Hiroshi Ishiguro. Reading Human Relationships from Their interaction with an interactive humanoid robot. In *Proceedings of International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 402–412, 2004.

[22] Takahiro Miyashita, Masahiro Shiomi, and Hiroshi Ishiguro. Multisensor-based Human Tracking Behaviors with Markov Chain Monte Carlo Methods. In *Proceedings of International Conference on Humanoid Robots*, 2004.

[23] Hiroshi G. Okuno, Kazuhiro Nakadai, Tino Lourens, and Hiroaki Kitano. Sound and Visual Tracking for Humanoid. In *Proceedings of 2000 International Conference on Information Society in the 21st Century: Emerging Technologies and New Challenges*, pages 254–261, 2000.

[24] Yosuke Matsusaka, Tsuyoshi Tojo, Sentaro Kubota, Kenji Furukawa, Daisuke Tamiya, Keisuke Hayata, Yuichiro Nakano, and Tetsunori Kobayashi. Multi-person

Conversation via Multi-modal Interface — A Robot who Communicates with Multi-user. In *Proceedings of European Conference on Speech Communication Technology*, pages 1723–1726, 1999.

[25] KyeongJu Kim, Yosuke Matsusaka, and Tetsunori Kobayashi. Inter-Module Cooperation Architecture for Interactive Robot. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 2286–2291, 2002.

[26] Zhen Jia, Arjuna Balasuriya, and Subhash Challa. Sensor Fusion based 3D Target Visual Tracking for Autonomous Vehicles with IMM. In *Proceedings of International Conference on Robotics and Automation*, pages 1841–1846, 2005.

[27] Takafumi Sonoura, Takashi Yoshimi, Manabu Nishiyama, Hideichi Nakamoto, Seiji Tokura, and Nobuto Matsuhira. Person Following Robot with Vision-based and Sensor Fusion Tracking Algorithm. In *Computer Vision*, pages 519–538, 2008.

[28] Matteo Munaro, Filippo Bassoo, and Emanuele Menegatti. Tracking People with Groups with RGB-D Data. In *International Conference on Intelligent Robots and Systems*, pages 2101–2107, 2012.

[29] Licong Zhang, Jurgen Sturm, Daniel Cremers, and Dongheui Lee. Real-Time Human Motion Tracking using Multiple Depth Cameras. In *International Conference on Intelligent Robotics and Systems*, pages 2389–2395, 2012.

[30] Martin Weser, Daniel Westhoff, Markus Hiiser, and Jianwei Zhang. Multimodal People Tracking and Trajectory Prediction based on Leamed Generalized Motion Patterns. In *Proceedings of International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 541–546, 2006.

[31] Zhichao Chen and Stanley T. Birchfield. Person Following with a Mobile Robot Using Binocular Feature-Based Tracking. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 815–820, 2007.

[32] Hyuksecong Kwon, Youngrock Yoon, Jae Byung Park, and Avinash C. Kak. Person Tracking with a Mobile Robot using Two Uncalibrated Independently Moving Cameras. In *Proceedings of International Conference on Intelligent Robotics and Automation*, pages 2877–2883, 2005.

[33] Ryosuke Ozeki, Yoshifumi Minoura, Hironobu Fujiyoshi, Tokihiko Akita, and Toshiaki Kakinami. Vehicle Tracking using Cooperative Multiple Scalable Mean-Shift Trackers. In *Meeting on Image Recognition and Understanding*, pages 419–426, 2005.

[34] Peter Nordlund and Tomas Uhlin. Closing the loop: Pursuing a Moving Object by a Moving Observer. *Image and Vision*, 14(4):400–407, 1996.

[35] Boyoon Jung and Gaurav S. Sukhatme. Detecting Moving Objects using a Single Camera on a Mobile Robot in an Outdoor Environement. In *Proceedings of International Conference on Autonomous Systems*, pages 980–987, 2004.

[36] Giorgio Chivilo, Flavio Mezzaro, Antonio Sgorbissa, and Renato Zaccaria. Follow-the-Leader Behavior through Optical Flow Minimization. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 3182–3187, 2004.

[37] Maurizio Piaggio, Roberto Formaro, Alberto Piombo, Luca Sanna, and Renato Zaccaria. An Optical-Flow Person Following Behavior. In *Proceedings of IEEE ISIC/CIRNISAS Joint Conference*, pages 4078–4083, 1998.

[38] Abhijit Kundu, K Madhava Krishna, and Jayanthi Sivaswamy. Moving Object Detection by Multi-View Geometric Techniques from a Single Camera Mounted Robot. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 4306–4312, 2009.

[39] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *International Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[40] Tomoki Watanabe, Satoshi Ito, and Kentaro Yokoi. Co-occurrence Histograms of Oriented Gradients for Human Detection. *IPSJ Transactions on Computer Vision and Applications*, 2:39–47, 2010.

[41] Geraldo Silveira, Ezio Malis, and Patrick Rives. Real-time Robust Detection of Planar Regions in a Pair of Images. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 49–54, 2006.

[42] Yunting Pang, Qiang Huang, Weimin Zhang, Zhangfeng Hu, Altaf Hussain Rajpar, and Keijie Li. Real-time Object Tracking of a Robot Head Based on Multiple Visual

Cues Integration. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 686–691, 2006.

[43] Yoshiro Negishi, Jun Miura, and Yoshiaki Shirai. Calibration of Omnidirectional Stereo for Mobile Robots. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 2600–2605, 2004.

[44] Kazumasa Yamazawa, Yasushi Yagi, and Masahiko Yachida. Visual Navigation with Omnidirectional Image Sensor HyperOmni Vision. *Journal of the Institute of Electronics, Information and Communication Engineers*, J79-D-2(5):698–707, 1996.

[45] Ian Horswill. Visual Collision Avoidance by Segmentation. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 902–909, 1994.

[46] Nguyen Xuan Dao, Bum-Jae You, and Sang-Rok Oh. Visual navigation for indoor mobile robots Using a single camera. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 3389–3394, 2005.

[47] Luis-Felipe Posada, Krishna Kumar Narayanan, Frank Hoffmann, and Torsten Bertram. Floor segmentation of omnidirectional images for mobile robot visual navigation. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 804–809, 2010.

[48] Nyungsik Kim, Nak Youg Chong, Hyo-Sung Ahn, and Wonpil Yu. RFID-enabled Target Tracking and Following with a Mobile Robot Using Direction Finding Antennas. In *Proceedings of International Conference on Automation Science and Engineering*, pages 1014–1019, 2007.

[49] Wonpil Yu, Yu-Cheol Lee, Sunglok Choi, Heesung Chae, and Jihoon Jung. Automated Robotic Service in Large-Scale Exhibition Environments. In *Proceedings of International Conference on Automation Science and Engineering*, pages 1152–1157, 2012.

[50] Cory Hekimian-Williams, Brandon Grant, Xiuwen Liu, Zhenghao Zhang, and Piyush Kumar. Accurate Localization of RFID Tags Using Phase Difference. In *International Conference on RFID*, pages 89–96, 2010.

[51] Hirohiko Kawata, Toshihiro Mori, and Shinichi Yuta. Design and Realization of 2-Dimensional Optical Range Senser for Environment Recognition in Mobil Robots. *Jounal of Robotics and Mechatronics*, 17(2):116–120, 2005.

[52] Hirohiko Kawata, Akihisa Ohya, and Shinich Yuta. Development of Compact and Light-weight LRF Based Positioning Sensor for Mobile Robot Localization. In *Proceedings of International Conference on Advanced Robotics*, pages 1–6, 2009.

[53] Paul Besl and McKay D. Neil. A Method for Registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(14):239–256, 1992.

[54] Zhang Zhengyou. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 12(13):119–152, 1994.

[55] Masahiro Tomono. Efficient Global Scan Matching Using Saliency-based Scan Point Resampling. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 1433–1438, 2005.

[56] A.C.Murillo, Jose Jesus Guerrero, and Carlos Sagues. SURF features for efficient robot localization with omnidirectional images. In *Proceedings of International Conference on Robotics and Autmation*, pages 3901–3907, 2007.

[57] Pierre Lamon, Adriana Tapus, Etienne Glauser, Nicola Tomatis, and Roland Siegwart. Environmental Modeling with Fingerprint Sequences for Topological Global Localization. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 3781–3786, 2003.

[58] Roland Bunschoten and Ben J. A. Krose. Robust Scene Reconstruction From an Omnidirectional Vision System. *IEEE Transactions on Robotics*, 19(2):351–357, 2003.

[59] Jonathan Ventura and Tobias Hollerer. Wide-Area Scene Mapping for Mobile Visual Tracking. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 3–12, 2012.

[60] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[61] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[62] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[63] Sunghwan Ahn, Wan Kyun Chung, and Sang-Rok Oh. Construction of hybrid visual map for indoor slam. In *International Conference on Intelligent Robots and Systems*, pages 1695–1701, 2007.

[64] Bruce D. Lucas ad Takeo Kanade. An Interative Image Registration Technique with an Application to Stereo Vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[65] Siegfried Hochdorfer and Christian Schlegel. Landmark rating and selection according to localization coverage: Addressing the challenge of lifelong operation of SLAM in service robots. In *International Conference on Intelligent Robots and Systems*, pages 382–387, 2009.

[66] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007.

[67] Jonathan Ventura and Tobias Hollerer. Wide-Area Scene Mapping for Mobile Visual Tracking. In *International Symposium on Mixed and Augmented Reality*, pages 3–12, 2012.

[68] Satoru Satake, Takayuki Kanda, Dylan F. Glas, Michita Imai, Hiroshi Ishiguro, and Norihiro Hagita. How to Approach Humans? -Strategies for Social Robots to Initiate Interaction-. In *International Conference on Human Robot Interaction*, pages 109–116, 2009.

[69] Fumitaka Yamaoka, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. A Model of Proximity Control for information-Presenting Robots. *IEEE Transactions on Robotics*, 26(1):187–195, 2010.

[70] Edward T. Hall. *The Hidden Dimension*. Doubleday Publishing, 1966.

[71] Matthew J. Hertenstein and Dacher Keltner. Touch communicates distinct emotions. *the American Psychological Association*, 6(3):528–533, 2006.

[72] Susan J. Lederman and Roberta L. Klatzky. Haptic Recognition of Static and Dynamic Expressions of Emotion in the Live Face. *Association for Psychological Science*, 18(2):158–164, 2006.

[73] Hiroaki Kawamichi, Ryo Kitada, Kazufumi Yoshihara, Haruka K. Takahashi, and Norihiro Sadato. Activation of te reward system by joining hands with familiar person: an fMRI study. In *8th International Brain Research Organization World Congress of Neuroscience*, pages C368–C368, 2011.

[74] Akinobu Lee and Tatsuya Kawahara. Recent development of open-source speech recognition engine Julius. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 131–137, 2009.

[75] Hiroshi G. Okuno, Kazuhiro Nakadai, Ken ichi Hidai, Hiroshi Mizoguchi, and Hiroaki Kitano. Human-Robot Interaction Through Real-Time Auditory and Visual Multiple-Talker Tracking. In *International Conference on Intelligent Robots and Systems*, pages 1402–1409, 2001.

[76] Kazuhiro Nakadai, Hiroshi G. Okuno, and Hiroaki Kitano. Real-Time Sound Source Localization and Separation for Robot Audition. In *IEEE International Conference on Spoken Language Processing*, pages 193–196, 2002.

[77] Takuma Otsuka, Kazuhiro Nakadai, Tetsuya Ogata, and Hiroshi G. Okuno. Incremental Bayesian Audio-to-Score Alignment with Flexible Harmonic Structure Models. In *International Society for Musical Information Retrieval Conference*, pages 525–530, 2011.

[78] Keisuke Nakamura, Kazuhiro Nakadai, Futoshi Asano, Yuji Hasegawa, and Hiroshi Tsujino. Intelligent Sound Source Localization for Dynamic Environments. In *International Conference on Intelligent Robots and Systems*, pages 664–669, 2009.

[79] Ralph O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, AP-34(3):276–280, 1986.

[80] Futoshi Asano, Yoichi Motomura, Hideki Asoh, Takashi Yoshimura, Naoyuki Ichimura, and Satoshi Nakamura. Fusion of audio and video information for detecting speech events. In *Proceedings of International Conference on Information Fusion*, volume 1, pages 386–393, 2003.

[81] Futoshi Asano, Kiyoshi Yamamoto, Isao Hara, Jun Ogata, Takashi Yoshimura, Yoichi Motomura, Naoyuki Ichimura, and Hideki Asoh. Detection and Separation of Speech Event Using Audio and Video Information Fusion and Its Application to Robust Speech Interface. *EURASIP Journal on Applied Signal Processing 2004*, pages 1727–1738, 2004.

[82] Fasel I. and Movellan J.R. Comparison of neurally inspired face detection algorithms. *Proc. of International Conference on Artificial Neural Networks (ICANN 2002)*, pages 1395–1401, 2002.

[83] Takahiro Miyashita, Masahiro Shiomi, and Hiroshi Ishiguro. A Communiation Robot covered with Soft Sensor Skin. In *The 21th Annual Conference of the Robotics Society of Japan*, pages 1A23–1A23, 2003.

[84] Takayuki Kanda, Hiroshi Ishiguro, and Toru Ishida. Psychological analysis on human-robot interaction. In *International Conference on Intelligent Robotics and Automation*, pages 4166–4173, 2001.

[85] Hidenori Takeshima, Takashi Ida, and Toshimitsu Kaneko. Extracting Object Regions Using Locally Es-timated Probability Density Functions. In *Proceedings of IAPR Conference on Machine Vision Applications*, pages 465–468, 2007.

[86] Joe H Ward. Hierarchical Grouping to Optimize an Objective Function. *the American Statistical Association*, 58(301):236–244, 1963.

[87] Takayuki Kanda and Hiroshi Ishiguro. Friendship estimation model for social robots to understand human relationships. In *International Workshop on Robot and Human Communication*, pages 539–544, 2004.

[88] John A. Nelder and Roger Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[89] Shai Avidan. Support Vector Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1064–1072, 2004.

[90] Takahiro Okabe and Yoichi Sato. Support Vector Machines for Object Recognition under Varying Illumination Conditions. In *Asian Conference on Computer Vision*, pages 724–729, 2004.

[91] Massimiliano Pontil and Alessandro Verri. Support Vector Machines for 3D Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637–646, 1998.

[92] Masahito Sano, Tsuyoshi Takanose, and Akiko Numata. Transfer Robot Aimed at Use in Retail Stores. *Toshiba Review*, 64(1):48–51, 2009.

[93] Seiji Tokura, Tsuyoshi Tasaki, Sonoura Takafumi, Masahito Sano, Matsuhira Nobuto, and Kiyoshi Komoriya. Robotic Transportation System for Shopping Support Services. In *International Symposium on Robot and Human Interactive Communication*, pages TuBH3–TuBH3, 2009.

[94] Nobuto Matsuhira, Fumio Ozaki, Seiji Tokura, Takafumi Sonoura, Tsuyoshi Tasaki, Hideki Ogawa, Masahito Sano, Akiko Numata, Naohisa Hashimoto, and Kiyoshi Komoriya. Development of robotic transportation system - shopping support system collaborating with environmental cameras and mobile robots -. In *International Symposium on and German Conference on Robotics*, pages 1–6, 2010.

[95] Takafumi Sonoura, Seiji Tokura, Tsuyoshi Tasaki, Fumio Ozaki, and Nobuto Matsuhira. Reflective Collision Avoidance for Mobile Service Robot in person coexistence environment. *Journal of Robotics and Mechatronics*, 23(6):999–1011, 2011.

[96] Jean-Yves Bouguet. *Pyramidal Implenentation of the Lucas Kanade Feature Tracker*. Intel Corporation, Microprocessor Research Labs, 1997.

[97] Ryosuke Kawanishi, Atsushi Yamashita, and Toru Kaneko. Three-Dimensional Environment Model Construction from an Omnidirectional Image Sequence. *Journal of Robotics and Mechatronics*, 21(5):574–582, 2009.

[98] Noriaki Ando, Takashi Suehiro, Kosei Kitagaki, Tetsuo Kotoku, and Woo-Keun Yoon. RTMiddleware: Distributed Component Middleware for RT (Robot- Tech-

nology). In *International Conference on Intelligent Robots and Systems*, pages 3555–3560, 2005.